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

Для создания депозитного калькулятора была выбрана среда выполнения Eclipce;
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-

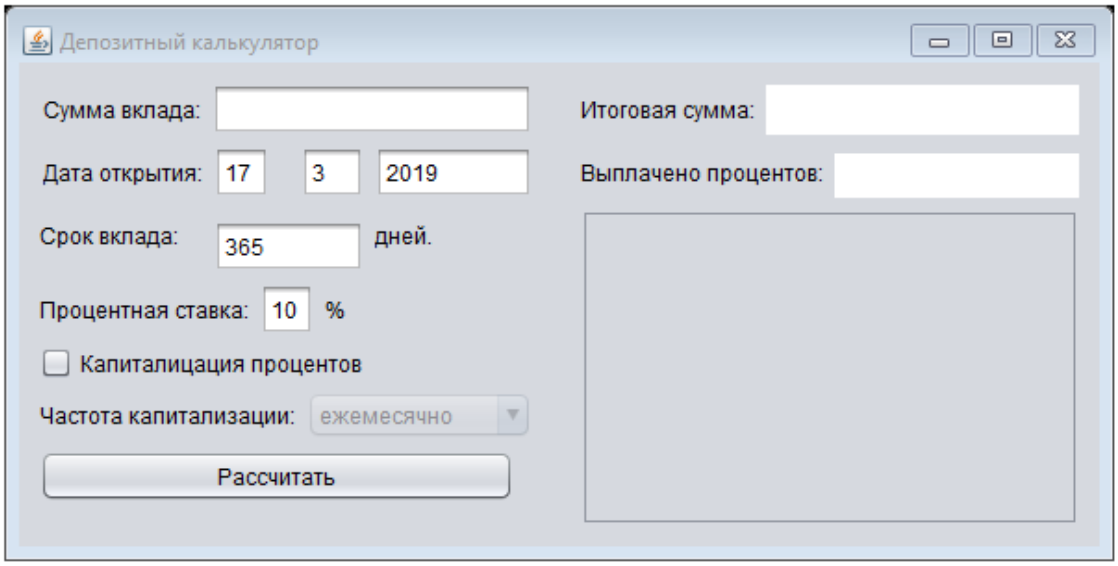

Рис. 1. Итоговый вид формы

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

Листинг 1. Нажатие на флажок

```
//ОБРАБОТКА ФЛАЖКА
ch_capital.addActionListener(new ActionListener() {
	public void actionPerformed(ActionEvent arg0) {
		if (ch_capital.isSelected()) //без капитализации вывод недоступен (т.к ничего не меняет)
		{c_type_proc.setEnabled(true);}else
		{c_type_proc.setEnabled(false);}
	}
});
```

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 энумератор 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 текстбоксов, 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

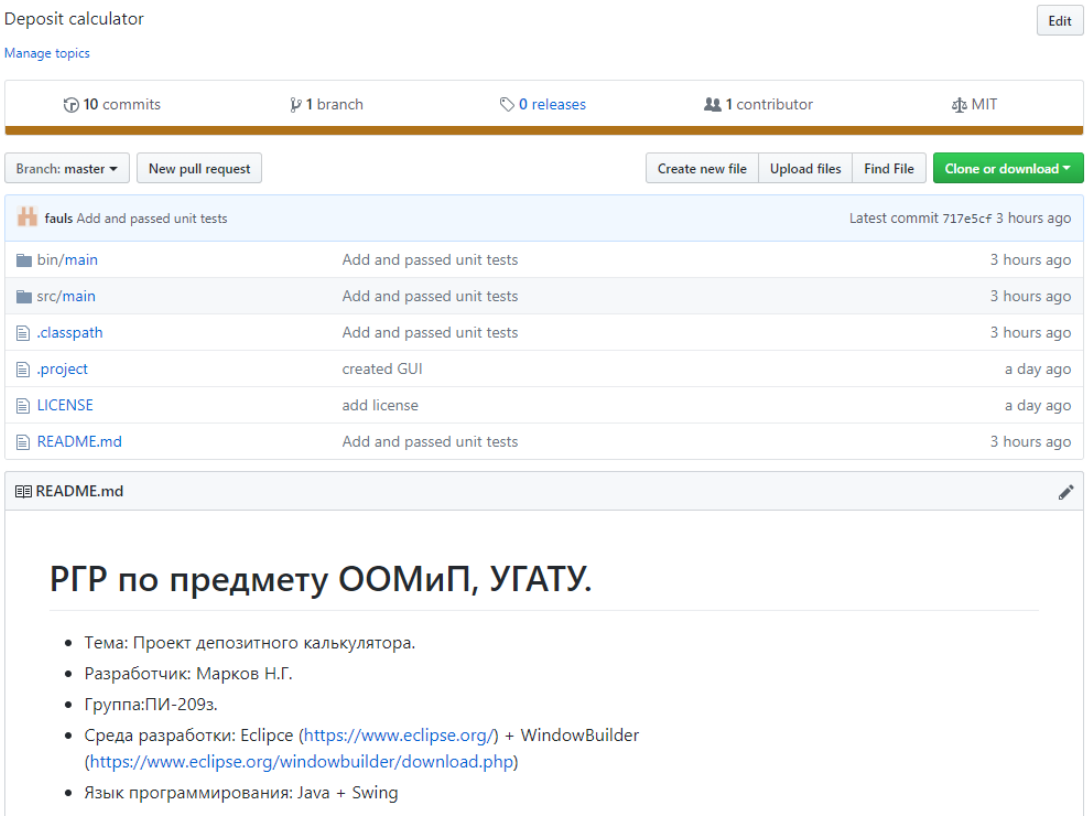

Рис. 2. Страница репозитория

3. Описание тестирования работы программы.

Для тестирования использовались JUnit5 тесты, содержащиеся в файле Test.java. В ходе тестирование 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.

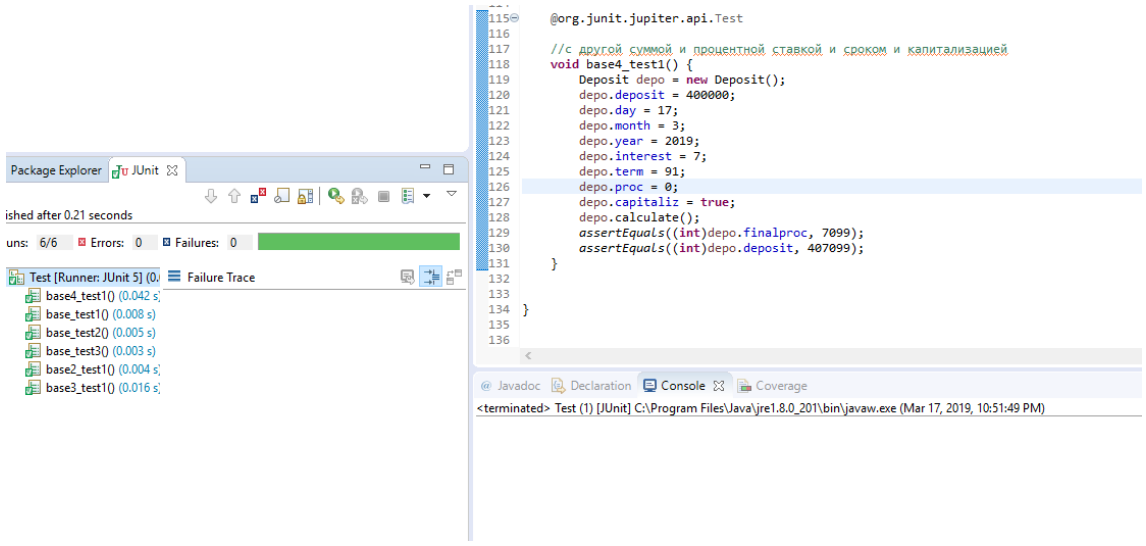

Рис. 3. Пример пройденного unit-тестирования

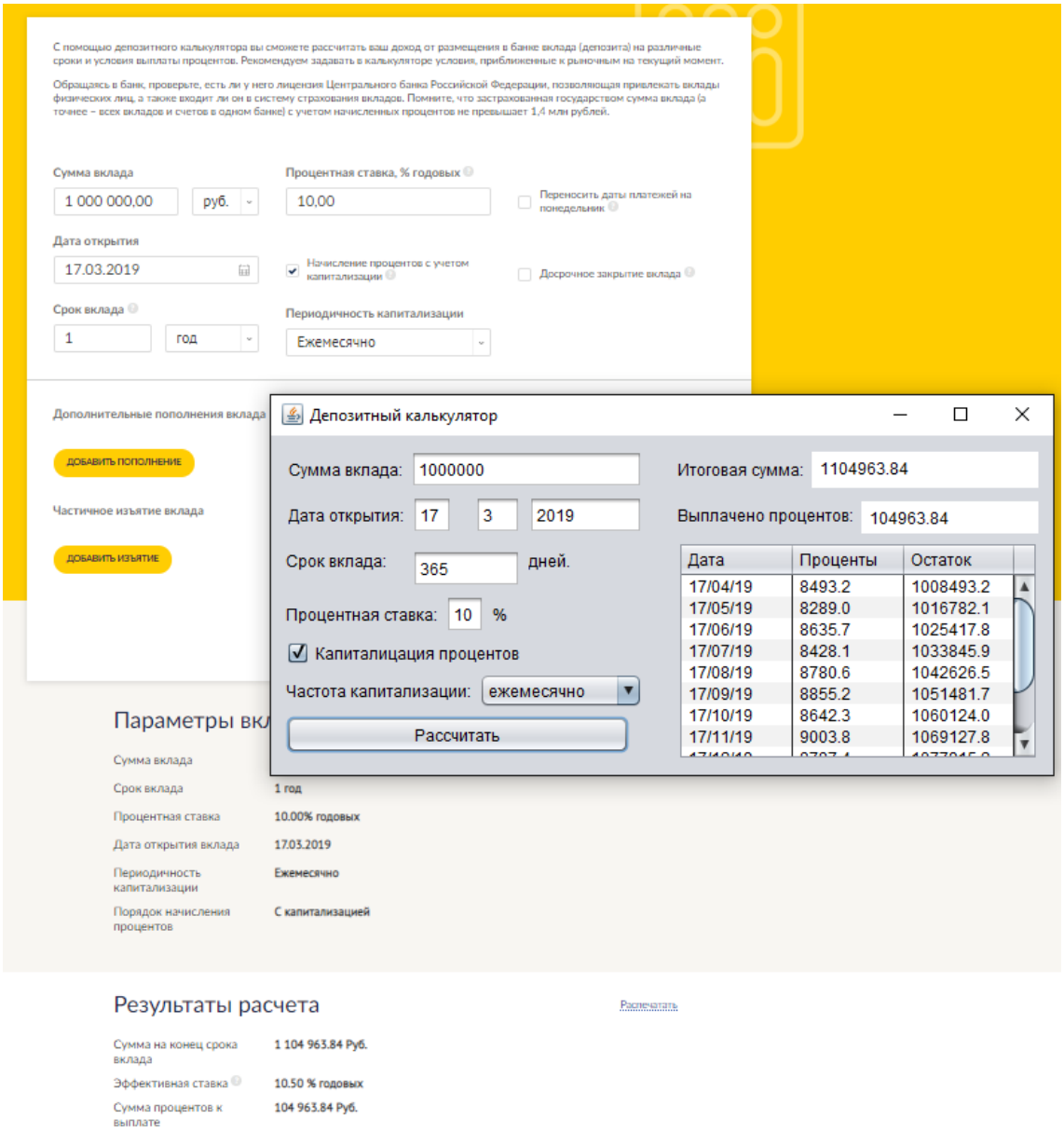

Рис. 4. Сравнение с онлайн-калькулятором

4. Приложение

4.1. Deposit.java - основной класс

Листинг 2. Основной класс

```
package main;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;

//
//основной класс для работы с депозитом
//
public class Deposit {
        double deposit; //сумма
        int year;
        int month;
        int day;
        int term; //срок
        double interest; //процентная ставка
        boolean capitaliz; //капитализация
        int proc =0; //0 - капитализация, 1 - через 3 месяца, 2 - ежегодно
        String[][] table; //таблица с датой, процентами и остатком
```

```
20
21          private LocalDate _startdate; //начальная дата
22          private LocalDate _stopdate; //конечная
23          private int _periods; //кол-во месяцев (проценты добавляются помесячно)
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
                        учитывается
41                  _periods = (int) ChronoUnit.MONTHS.between(_startdate, _stopdate);//кол-во полных месяцев в периоде
42          }
43
44          private double howDay()//сколько дней в месяце
45          {
46                  return _startdate.lengthOfMonth();
47          }
48
49          private double mathMoney()//подсчёт по формуле
50          {
51                  return deposit*((interest/100)*(howDay()/_startdate.lengthOfYear())); //подневный расчет
52          }
53
54          private void addCapital(double capMoney, int i)//добавить сумму к финальной, внести строку в таблицу,
                сдвинуть дату
55          {
56                  finalproc += capMoney; //финальные проценты - сумма всех процентов
57                  switch(proc)//добавление месяцев, в зависимости от частоты вывода в таблицу
58                  {
59                  case 0: _startdate = _startdate.plusMonths(1); break;//добавление месяца
60                  case 1: _startdate = _startdate.plusMonths(3); break;
61                  case 2: _startdate = _startdate.plusYears(1); break;
62                  }
63                  table[i][0]=_startdate.format(DateTimeFormatter.ofPattern("d/MM/YY"));//форматированный вывод даты в
                        первую колонку
64                  table[i][1]=String.format("%.1f",capMoney); //формат до 2х знаков после запятой
65                  table[i][2]=String.format("%.1f",deposit);
66          }
67
68          private void simpleInterest()//простые проценты, для них выдача/невыдача денег не важна, так что proc
                игнорируется
69          {
70
71                  table = new String[_periods][3]; //инициализация таблицы
72
73                  for(int i=0; i<_periods;i++) //добавление процентов каждый период
74                  {addCapital(mathMoney(), i); //считаем
75                  }
76                  deposit+=finalproc; //к итогу добавляем проценты
77          }
78
79          private void hardInterest()//сложные проценты
80          {
81                  table = new String[_periods][3]; //инициализация таблицы
82                  double capMoney = 0; //накопление процентов за период
83                  int clock = 0; //для отсчёта периодов
84                  int period =0; //для ровного вывода строк
85                  for(int i=0; i<_periods;i++) //добавление процентов каждый период
86                  {
87                          capMoney+=mathMoney(); //накапливать за период
88                          clock++; //отсчёт периодов
89
90                          switch(proc)//проведение капитализации в зависимости от выбранного варианта
91                          {
92                          case 0: if (clock==1)  {deposit +=capMoney; addCapital(capMoney, period); period++; capMoney
                                =0; clock=0;} break;
```

```
93                      case 1: if (clock==3)  {deposit +=capMoney; addCapital(capMoney, period); period++; capMoney
                             =0; clock=0;} break;
94                      case 2: if (clock==12)  {deposit +=capMoney; addCapital(capMoney, period); period++; capMoney
                             =0; clock=0;} break;
95                      }
96
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 - энумератор

Листинг 4. ENUM для текстбокса

```
package main;
//энумератор для комбобокса
public enum Proc {
	ежемесячно,
	ежеквартально,
	ежегодно
}
```

4.4. wind.java - GUI и ввод

Листинг 5. GUI и обработка ввода

```
package main;

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.UIManager;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;

import java.time.LocalDate;
import java.time.Month;
import javax.swing.JCheckBox;
import javax.swing.JButton;
import javax.swing.JTextPane;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.JScrollPane;

public class wind extends JFrame {

	private JPanel contentPane;
	private JTextField t_sum;
	private JTextField t_day;
	private JTextField t_year;
	private JTextField t_month;
	private JLabel l_time_vklad;
	private JTextField t_time_vklad;
	private JLabel label;
	private JLabel l_proc;
	private JTextField t_proc;
	private JLabel label_2;
	private JLabel label_3;
	private JLabel label_4;
	private JButton b_go;
	private JCheckBox ch_capital;
	private JTable table;
	private JTextPane t_prok_out;
	private JTextPane t_ost_vklad;
	private JComboBox c_type_proc;

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		try {
			UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");
		} catch (Throwable e) {
			e.printStackTrace();
		}
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					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                          {c_type_proc.setEnabled(true);}else
91                          {c_type_proc.setEnabled(false);}
92                  }
93          });
94
95          //ОБРАБОТКА КНОПКИ
96          b_go.addActionListener(new ActionListener() {
97
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
                                        исключение если выходим за пределы
116                                 {deposit.day = day;}
117                                 } catch (NumberFormatException e) {
118                                         JOptionPane.showMessageDialog(null, "Введите␣день␣числом␣от␣1␣до␣31");
119                                 }
120                         try{//месяц
121                                 int month = Integer.parseInt(t_month.getText());
122
123                                 if (month<1 || month>12) {throw new NumberFormatException();}else
124                                 {deposit.month = month;}
125
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
135                                 } catch (NumberFormatException e) {
136                                         JOptionPane.showMessageDialog(null, "Введите␣год␣числом␣больше␣0");
137                                 }
138
139                         try{//срок вклада
140                                 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,//двумерным 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\u043F\u043E\u0437\u0438\u0442\u043D\u044B\u0439␣\u043A\u0430\u043B\u044C\u043A
                    \u0443\u043B\u044F\u0442\u043E\u0440");
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␣\u0432\u043A\u043B\u0430\u0434\u0430:");
200
201             t_sum = new JTextField();
202             t_sum.setToolTipText("\u0412\u0432\u0435\u0434\u0438\u0442\u0435␣\u0441\u0443\u043C\u043C\u0443");
203             t_sum.setColumns(10);
204
205             JLabel l_date = new JLabel("\u0414\u0430\u0442\u0430␣\u043E\u0442\u043A\u0440\u044B\u0442\u0438\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     //ввод текущей даты в текстбоксы
215             t_day = new JTextField();
216             t_day.setText(""+day);
```

```
217                 t_day.setColumns(10);
218
219                 t_year = new JTextField();
220                 t_year.setText(""+year);
221                 t_year.setColumns(10);
222
223                 t_month = new JTextField();
224                 t_month.setText(""+(month.ordinal()+1));
225                 t_month.setColumns(10);
226
227                 //автогенерация
228
229                 //Hic sunt dracones
230                 l_time_vklad = new JLabel("\u0421\u0440\u043E\u043A␣\u0432\u043A\u043B\u0430\u0434\u0430:");
231
232                 t_time_vklad = new JTextField();
233                 t_time_vklad.setToolTipText("\u0412␣\u0434\u043D\u044F\u0445");
234                 t_time_vklad.setText("365");
235                 t_time_vklad.setColumns(10);
236
237                 label = new JLabel("\u0434\u043D\u0435\u0439.");
238
239                 l_proc = new JLabel("\u041F\u0440\u043E\u0446\u0435\u043D\u0442\u043D\u0430\u044F␣\u0441\u0442\u0430\
                        u0432\u043A\u0430:");
240
241                 t_proc = new JTextField();
242                 t_proc.setToolTipText("\u0412␣\u043F\u0440\u043E\u0446\u0435\u043D\u0442\u0430\u0445");
243                 t_proc.setText("10");
244                 t_proc.setColumns(10);
245
246                 label_2 = new JLabel("%");
247
248                 ch_capital = new JCheckBox("\u041A\u0430\u043F\u0438\u0442\u0430\u043B\u0438\u0446\u0430\u0446\u0438\
                        u044F␣\u043F\u0440\u043E\u0446\u0435\u043D\u0442\u043E\u0432");
249
250                 ch_capital.setToolTipText("\u041F\u0440\u043E\u0446\u0435\u043D\u0442\u044B␣\u0441␣\u0443\u0441\u0442\
                        u0430\u043D\u043E\u0432\u043B\u0435\u043D\u043D\u043E\u0439␣\u043F\u0435\u0440\u0435\u043E\
                        u0434\u0438\u0447\u043D\u043E\u0441\u0442\u044C\u044E␣\u0434\u043E\u0431\u0430\u0432\u043B\u044F
                        \u044E\u0442\u0441\u044F␣\u043A␣\u0432\u043A\u043B\u0430\u0434\u0443");
251
252                 JLabel label_1 = new JLabel("Частота␣капитализации:");
253
254                 c_type_proc = new JComboBox();
255                 c_type_proc.setToolTipText("\u0412\u044B\u043F\u043B\u0430\u0442\u0430␣\u0438\u043C\u0435\u0435\u0442
                        ␣\u0437\u043D\u0430\u0447\u0435\u043D\u0438\u0435␣\u0442\u043E\u043B\u044C\u043A\u043E␣\u043F\
                        u0440\u0438␣\u043A\u0430\u043F\u0438\u0442\u0430\u043B\u0438\u0437\u0430\u0446\u0438\u0438␣\
                        u043F\u0440\u043E\u0446\u0435\u043D\u0442\u043E\u0432");
256                 c_type_proc.setEnabled(false);
257                 c_type_proc.setModel(new DefaultComboBoxModel(Proc.values()));
258
259                 b_go = new JButton("\u0420\u0430\u0441\u0441\u0447\u0438\u0442\u0430\u0442\u044C");
260
261
262                 label_3 = new JLabel("Итоговая␣сумма:");
263
264                 label_4 = new JLabel("\u0412\u044B\u043F\u043B\u0430\u0447\u0435\u043D\u043E␣\u043F\u0440\u043E\u0446
                        \u0435\u043D\u0442\u043E\u0432:");
265
266                 t_ost_vklad = new JTextPane();
267                 t_ost_vklad.setEditable(false);
268
269                 t_prok_out = new JTextPane();
270                 t_prok_out.setEditable(false);
271
272                 JScrollPane scrollPane = new JScrollPane();
273
274                 GroupLayout gl_contentPane = new GroupLayout(contentPane);
275                 gl_contentPane.setHorizontalGroup(
276                         gl_contentPane.createParallelGroup(Alignment.LEADING)
277                                 .addGroup(gl_contentPane.createSequentialGroup()
278                                         .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
279                                                 .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
280                                                         .addGroup(gl_contentPane.createSequentialGroup()
281                                                                 .addContainerGap()
282                                                                 .addComponent(l_time_vklad, GroupLayout.PREFERRED_SIZE,
                                                                         94, GroupLayout.PREFERRED_SIZE))
283                                                         .addGroup(gl_contentPane.createSequentialGroup()
284                                                                 .addGap(8)
```

```
                                                .addGroup(gl_contentPane.createParallelGroup(Alignment.
                                                      LEADING)
                                                  .addGroup(gl_contentPane.createSequentialGroup()
                                                        .addComponent(l_sum)
                                                        .addPreferredGap(ComponentPlacement.RELATED)
                                                        .addComponent(t_sum, GroupLayout.
                                                              DEFAULT_SIZE, 151, Short.MAX_VALUE))
                                                  .addGroup(gl_contentPane.createSequentialGroup()
                                                        .addComponent(l_date)
                                                        .addPreferredGap(ComponentPlacement.RELATED)
                                                        .addGroup(gl_contentPane.createParallelGroup
                                                              (Alignment.LEADING)
                                                            .addGroup(gl_contentPane.
                                                                  createSequentialGroup()
                                                              .addComponent(t_day,
                                                                    GroupLayout.
                                                                    PREFERRED_SIZE, 30,
                                                                    GroupLayout.
                                                                    PREFERRED_SIZE)
                                                              .addPreferredGap(
                                                                    ComponentPlacement.
                                                                    RELATED, GroupLayout.
                                                                    DEFAULT_SIZE, Short.
                                                                    MAX_VALUE)
                                                              .addComponent(t_month,
                                                                    GroupLayout.
                                                                    PREFERRED_SIZE, 34,
                                                                    GroupLayout.
                                                                    PREFERRED_SIZE))
                                                            .addComponent(t_time_vklad, 0, 0,
                                                                  Short.MAX_VALUE))
                                                        .addPreferredGap(ComponentPlacement.RELATED)
                                                        .addGroup(gl_contentPane.createParallelGroup
                                                              (Alignment.LEADING)
                                                            .addComponent(label)
                                                            .addComponent(t_year, GroupLayout.
                                                                  DEFAULT_SIZE, 74, Short.
                                                                  MAX_VALUE)))))
                                        .addGroup(gl_contentPane.createSequentialGroup()
                                                .addContainerGap()
                                                .addComponent(l_proc)
                                                .addPreferredGap(ComponentPlacement.RELATED)
                                                .addComponent(t_proc, GroupLayout.PREFERRED_SIZE, 29,
                                                      GroupLayout.PREFERRED_SIZE)
                                                .addPreferredGap(ComponentPlacement.RELATED)
                                                .addComponent(label_2))
                                        .addGroup(gl_contentPane.createSequentialGroup()
                                                .addContainerGap()
                                                .addComponent(ch_capital))
                                        .addGroup(gl_contentPane.createSequentialGroup()
                                                .addContainerGap()
                                                .addComponent(label_1)
                                                .addPreferredGap(ComponentPlacement.RELATED)
                                                .addComponent(c_type_proc, GroupLayout.PREFERRED_SIZE,
                                                      GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                ))
                                .addGroup(gl_contentPane.createSequentialGroup()
                                        .addContainerGap()
                                        .addComponent(b_go, GroupLayout.DEFAULT_SIZE, 266, Short.
                                              MAX_VALUE)
                                        .addGap(18)))
                        .addGap(18)
                        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 277, Short.MAX_VALUE)
                                .addGroup(gl_contentPane.createSequentialGroup()
                                        .addComponent(label_3)
                                        .addPreferredGap(ComponentPlacement.RELATED)
                                        .addComponent(t_ost_vklad, GroupLayout.DEFAULT_SIZE, 180, Short.
                                              MAX_VALUE))
                                .addGroup(gl_contentPane.createSequentialGroup()
                                        .addComponent(label_4)
                                        .addPreferredGap(ComponentPlacement.RELATED)
                                        .addComponent(t_prok_out, GroupLayout.DEFAULT_SIZE, 140, Short.
                                              MAX_VALUE)))
                        .addContainerGap())
        );
        gl_contentPane.setVerticalGroup(
                gl_contentPane.createParallelGroup(Alignment.LEADING)
                        .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.
                                                             DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
343                                                     .addComponent(label_3))
344                                             .addComponent(t_ost_vklad, GroupLayout.PREFERRED_SIZE, 27, GroupLayout.
                                                     PREFERRED_SIZE))
345                                     .addPreferredGap(ComponentPlacement.RELATED)
346                                     .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING)
347                                             .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
348                                                     .addComponent(l_date)
349                                                     .addComponent(t_day, GroupLayout.PREFERRED_SIZE, GroupLayout.
                                                             DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
350                                                     .addComponent(t_year, GroupLayout.PREFERRED_SIZE, GroupLayout.
                                                             DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
351                                                     .addComponent(t_month, GroupLayout.PREFERRED_SIZE, GroupLayout.
                                                             DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
352                                                     .addComponent(label_4))
353                                             .addComponent(t_prok_out, GroupLayout.PREFERRED_SIZE, GroupLayout.
                                                     DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
354                                     .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
355                                             .addGroup(gl_contentPane.createSequentialGroup()
356                                                     .addGap(12)
357                                                     .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
358                                                             .addGroup(gl_contentPane.createSequentialGroup()
359                                                                     .addGroup(gl_contentPane.createParallelGroup(
                                                                             Alignment.BASELINE)
360                                                                       .addComponent(l_time_vklad)
361                                                                       .addComponent(t_time_vklad, GroupLayout.
                                                                             PREFERRED_SIZE, GroupLayout.
                                                                             DEFAULT_SIZE, GroupLayout.
                                                                             PREFERRED_SIZE))
362                                                                     .addPreferredGap(ComponentPlacement.RELATED)
363                                                                     .addGroup(gl_contentPane.createParallelGroup(
                                                                             Alignment.BASELINE)
364                                                                       .addComponent(l_proc)
365                                                                       .addComponent(t_proc, GroupLayout.
                                                                             PREFERRED_SIZE, GroupLayout.
                                                                             DEFAULT_SIZE, GroupLayout.
                                                                             PREFERRED_SIZE)
366                                                                       .addComponent(label_2)))
367                                                             .addComponent(label))
368                                                     .addPreferredGap(ComponentPlacement.RELATED)
369                                                     .addComponent(ch_capital)
370                                                     .addPreferredGap(ComponentPlacement.RELATED)
371                                                     .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
372                                                             .addComponent(label_1)
373                                                             .addComponent(c_type_proc, GroupLayout.PREFERRED_SIZE,
                                                                     GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                                     )
374                                                     .addPreferredGap(ComponentPlacement.RELATED)
375                                                     .addComponent(b_go))
376                                             .addGroup(gl_contentPane.createSequentialGroup()
377                                                     .addPreferredGap(ComponentPlacement.RELATED)
378                                                     .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 171, Short.
                                                             MAX_VALUE)
379                                                     .addContainerGap())))
380             );
381
382             table = new JTable();
383             scrollPane.setViewportView(table);
384             table.setShowVerticalLines(true);
385             table.setRowSelectionAllowed(false);
386             contentPane.setLayout(gl_contentPane);
387     }
388 }
```