

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

# 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
5             недоступен (т.к ничего не меняет)
6             {c_type_proc.setEnabled(true);} else
7             {c_type_proc.setEnabled(false);}
8     }
9 });

```

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>

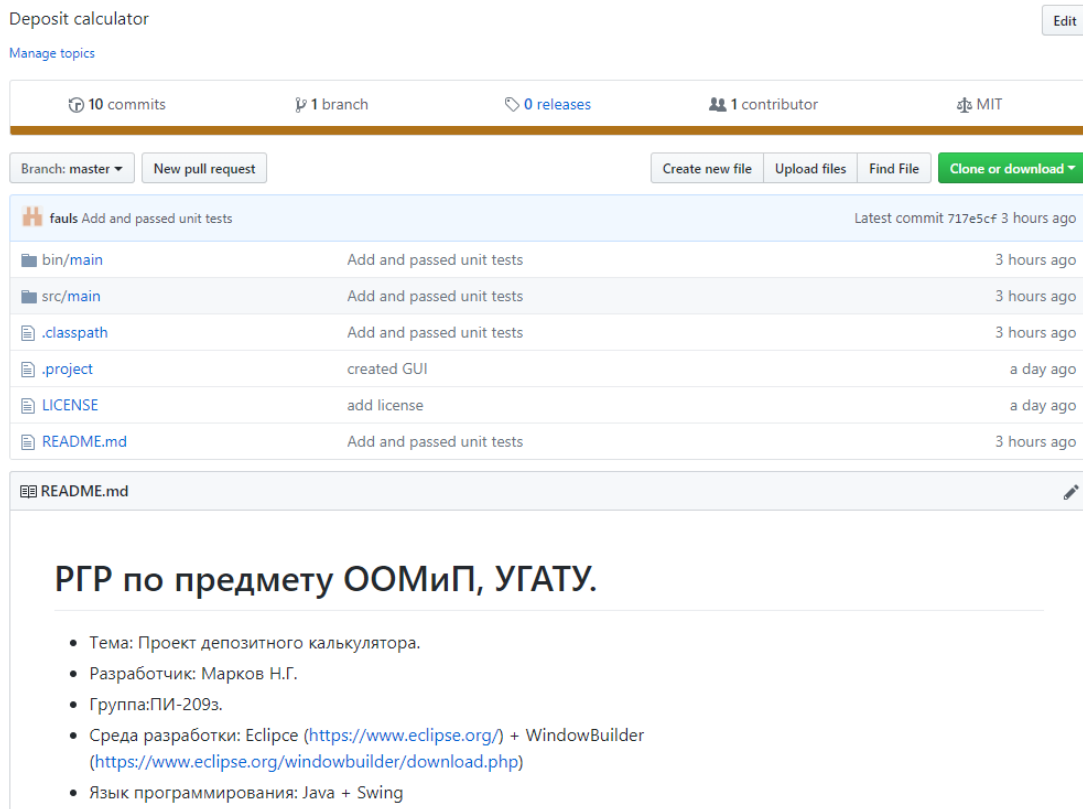


Рис. 2. Страница репозитория

### 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.

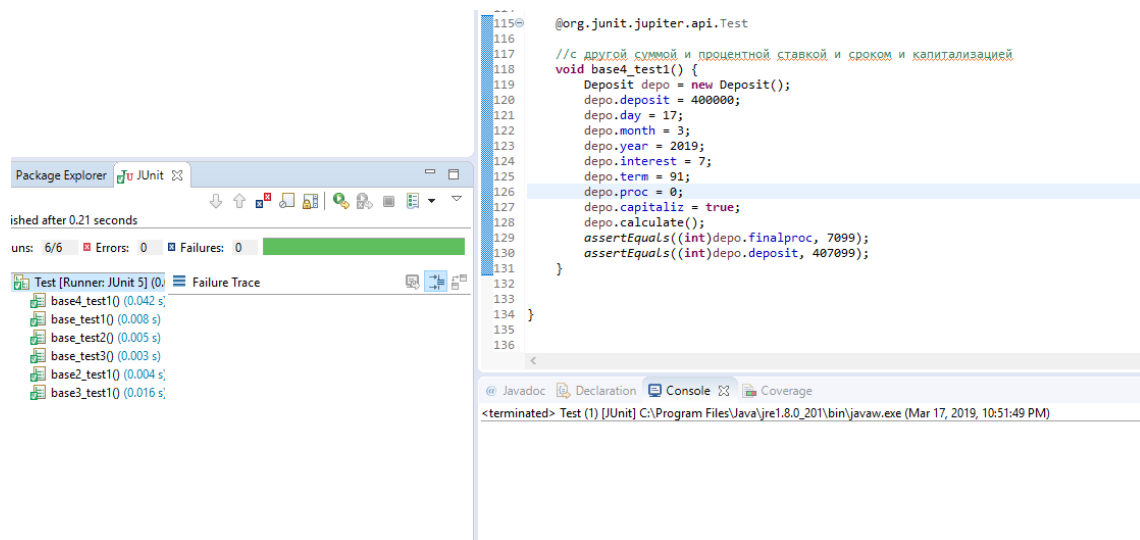


Рис. 3. Пример пройденного unit-тестирования