

Автономная некоммерческая организация дополнительного
профессионального образования
«Открытый молодёжный университет»

**КОМПЛЕКСНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
«ШКОЛЬНЫЙ УНИВЕРСИТЕТ»**

УТВЕРЖДАЮ

Директор
Автономной некоммерческой
организации дополнительного
профессионального образования
«Открытый молодёжный университет»

_____ И. В. Дмитриев
« ___ » _____ 2017 г.

Современное программирование на Java

Задание к итоговому проекту

Кузнецов В. В. Современное программирование на Java : Задание к итоговому проекту. — Томск : АНО ДПО «Открытый молодёжный университет», 2017. — 12 с.

Задание к итоговому проекту рассмотрено и рекомендовано к изданию на методическом семинаре Автономной некоммерческой организации дополнительного профессионального образования «Открытый молодёжный университет» 26 января 2017 года.

ЗАДАНИЯ К ИТОГОВОМУ ПРОЕКТУ

Задание 1. Калькулятор

Реализовать приложение калькулятора. Приложение должно иметь графический интерфейс. Калькулятор должен содержать в своём интерфейсе:

- кнопки от 0 до 9;
- кнопки стандартных операций: вычитание, сложение, умножение, деление;
- кнопку знака равенства;
- кнопки задания дробной части и знака минуса;
- кнопку сброса результата.

Кроме того, калькулятор должен содержать кнопки специальных операций, которые вместо надписей должны иметь соответствующие графические пиктограммы:

- извлечение квадратного корня;
- возведение числа в произвольную степень;
- расчёт значения $1/x$.

Принцип работы калькулятора должен быть таким же, как и у стандартных калькуляторов, т.е. вначале вводится первое число, затем операция, а затем второе число, и нажимается кнопка равенства.

Примечание: ввод второго числа и нажатие на кнопку равенства необходимо, если в расчёте участвуют два операнда.

В коде программы должен быть реализован отдельный класс, который реализует операции над числами. Вычисление результата должно быть основано на вызове методов этого класса.

Приложение должно иметь кнопку **О программе**, которая вызывает одноимённое модальное диалоговое окно, в котором содержится информация о разработчиках и предназначении приложения.

Задание 2. Текстовый редактор

Реализовать графическое приложение текстового редактора. Приложение, кроме компонента многострочного редактора, должно содержать кнопки:

- **Новый документ** — очистка содержимого окна текстового редактора.
- **Открыть документ** — открыть текстовый файл с диска, используя стандартный диалог для открытия файлов.
- **Сохранить документ** — сохранить открытый текстовый файл на диск под тем же самым именем, если файл существует на диске (был до этого момента открыт

или сохранён), в противном случае действие этой кнопки аналогично действию кнопки **Сохранить документ как...**

- **Сохранить документ как...** — сохранить текстовый документ на диск, используя стандартный диалог для сохранения файлов.
- **О программе** — вызвать одноимённое модальное диалоговое окно, в котором содержится информация о разработчиках и предназначении приложения.
- **Выход из программы.**

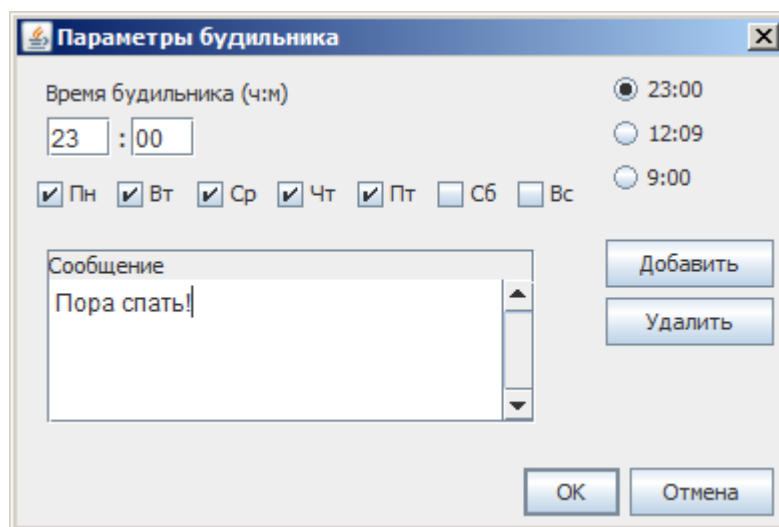
Кнопки вместо надписей должны иметь соответствующие пиктограммы.

Если текстовый файл был открыт с диска или сохранён на диск, то в заголовке программы должен отображаться полный путь к этому файлу.

В коде программы должен быть реализован отдельный класс, который реализует операции работы с файлом. Открытие файла и сохранение в файл данных должны происходить через вызов методов этого класса.

Задание 3. Часы со списком будильников

Создай графическое приложение будильника. Кроме отображения времени, даты и дня недели, как это было реализовано в уроках 23 и 24, часы должны хранить список времени срабатывания будильников. Этот список должен отображаться в отдельном диалоговом окне, например, таком:



Список времени будильников представлен в виде зависимых переключателей. Список изменяется динамически при помощи кнопок **Добавить** и **Удалить**, т.е. при нажатии кнопки **Добавить** появляется новый элемент списка — зависимый переключатель, а при нажатии кнопки **Удалить** выбранный будильник удаляется.

Кроме времени срабатывания и сообщения, отображаемого при срабатывании будильника, должны задаваться дни недели, когда будильник должен сработать, при помощи независимых переключателей.

Список будильника должен храниться в файлах: каждый будильник — это отдельный файл. Папка хранения файла — это папка с приложением. Сохранение параметров будильников в файлы должно происходить при нажатии на кнопку **ОК** диалога, а загрузка данных будильника должна происходить при запуске приложения.

В коде программы должен быть реализован отдельный класс, который реализует хранение данных отдельного будильника, а также класс, который реализует методы списка будильников. Последний должен иметь методы синхронизации параметров будильника с файлами.

Приложение должно иметь кнопку **О программе**, которая вызывает одноимённое модальное диалоговое окно, в котором содержится информация о разработчиках и предназначении приложения.

Задание 4. Игра на внимательность

Создай графическое приложение игры, которая тренирует внимательность игрока.

Суть игры: по полю случайным образом разбросаны квадраты двух цветов, которые случайным образом перемещаются по полю (имитируя броуновское движение), но не выходят за его пределы. Пользователь должен при помощи мыши убрать с экрана квадраты только одного, «разрешённого» цвета, т.е. при щелчке по «разрешённому» квадрату, он должен удаляться с экрана. Если пользователь щёлкает по квадрату с «запрещённым» цветом, то игра заканчивается. В интерфейсе также должно отображаться время игры. Основное окно игры может выглядеть следующим образом:



Игра должна иметь кнопку **Новая игра**, при нажатии на которую открывается диалог параметров игры:

- Задание цвета «выигрышных» и «проигрышных» квадратов.
- Задание размера квадратов: маленькие, средние, большие.
- Задание скорости движения квадратов в единицах: число пикселей за 100 мсек.

Этот диалог должен иметь кнопку **Начать игру**, при нажатии на которую диалог закрывается и начинается игра.

Дополнительно в игре может быть реализовано сохранение статистики в файл по каждой выигранной игре. Файл должен находиться в том же каталоге, что и приложение. В случае победы, запрашивается имя игрока и в файл заносится запись о времени игры, а также о всех её параметрах. При реализации этой возможности в интерфейс приложения необходимо добавить кнопку **Просмотр статистики**, при нажатии на которую будет отображаться отдельное окно, где будет размещён табличный компонент с данными о игре. Таблица должна иметь поля:

Имя	Время	Размер	Скорость	Цвет выигр.	Цвет проигр.
-----	-------	--------	----------	-------------	--------------

Приложение должно иметь кнопку **О программе**, которая вызывает одноимённое модальное диалоговое окно, в котором содержится информация о разработчиках и предназначении приложения.

Задание 5. Модернизация и рефакторинг приложения графического редактора

Произвести рефакторинг кода и добавить новый функционал в приложение рисования из проекта **Lesson21**, который выполнялся в течении урока 21 и 22.

Новый функционал

Добавить возможность сохранять расположение, размер и цвет всех нарисованных на холсте квадратов в файл, а также считывать эти данные из файла. После считывания холст должен очищаться и заполняться новыми квадратами на основе данных из открываемого файла. Формат хранения данных в файле определи самостоятельно и отрази в отчёте.

Рефакторинг

В коде программы должен быть реализован отдельный класс, который описывает отдельный квадрат (положение, размер и цвет), назовём его **Square**. А также класс, который содержит список объектов класса **Square** и методы для сохранения этого списка на диск, а также для загрузки данных из файла и формирования списка объектов класса **Square**. Основной код программы должен быть переписан с использованием методов этих двух классов.

Приложение должно иметь кнопку **О программе**, которая вызывает одноимённое модальное диалоговое окно, в котором содержится информация о разработчиках и предназначении приложения.

Задание 6. Свободная тема (п согласованию с учителем!)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

С помощью предлагаемых заданий можно оценить уровень усвоения материала по учебному курсу «Современное программирование на Java». Задание предполагает реализацию итогового проекта (приложения).

Цели итогового проекта:

- закрепление навыков работы в среде Eclipse: написание и отладка кода;
- закрепление навыков работы с простыми и ссылочными типами данных;
- закрепление навыков работы с операторами условий и циклов;
- эффективное использование ООП при конструировании логики приложений;
- закрепление на практике использования стандартных классов Java по работе со строками, датой и временем, математическими выражениями, динамическими массивами, файлами и т.д.;
- эффективное использование «ловушек» исключений в коде программы;
- овладение общими принципами создания консольных и оконных приложений с использованием библиотеки SWING;
- создание интерактивных приложений на основе событийной модели слушателей компонентов SWING.

Задание к итоговому проекту учащиеся выбирают на последнем практическом занятии по курсу. Задания могут быть выбраны из предложенных либо могут быть разработаны самостоятельно и согласованы с учителем.

Оценка работ ведётся по балльной шкале; при необходимости можно самостоятельно разработать более сложную систему, рассчитанную на ученическое жюри.

Система оценивания

Приложение оценивается по внешнему виду (дизайну) и функциональности, а также по правильности оформления кода: код должен легко читаться и быть снабжён комментариями.

Особое внимание должно уделяться использованию принципов ООП. Учащиеся должны сформулировать сущности проекта, которые могут быть реализованы в классах, и реализовать эти классы в коде. Также оценивается презентация проекта и пояснительная записка к нему.

Критерии оценивания	Количество баллов
Презентация проекта во время публичной защиты	3
Пояснительная записка к проекту, выполненная в соответствии с шаблоном	2

Критерии оценивания	Количество баллов
Соответствие функционала приложения техническому заданию	2
Дизайн приложения — лаконичность, удобство, привлекательность	3
Техническое исполнение — использование принципов ООП при реализации поставленной задачи. Грамотность и полнота реализации классов	5
Использование при разработке дополнительных библиотек компонентов, не рассматриваемых в курсе	3
Степень сложности реализации проекта	5
Наличие подробных комментариев к программному коду	2

Итоговая оценка выставляется согласно таблице:

Общее количество баллов	Оценка
Менее 7	2
7–12	3
13–17	4
18–25	5

Структура пояснительной записки к проекту

- **Титульный лист** (см. Приложение А) является первой страницей отчёта и содержит следующие сведения:
 - наименование организации;
 - наименование работы;
 - тему проекта;
 - данные об исполнителях;
 - данные о руководителе проектной деятельности;
 - место и дату составления отчёта.
- **Техническое задание** (см. Приложение Б) содержит данные, требуемые для разработки проекта, и включает в себя следующую информацию:
 - данные об утверждении темы (Ф. И. О. менеджера проекта, дата утверждения);
 - данные об исполнителях;
 - тему проекта;
 - срок исполнения проекта по этапам;
 - цели проекта;
 - задачи.
- **Аннотация** содержит краткую характеристику проекта, тему, проблему. Можно указать аудиторию, которой данная работа будет интересна и полезна.
- **Оглавление.**
- **Введение** требует краткого отражения следующих пунктов:
 - постановка задачи;
 - указание цели работы;
 - определение основной темы работы и подчинение ей частных задач;
 - этапы реализации проекта.
- **Содержательная часть** должна иметь следующие подразделы:
 - этапы разработки приложения;
 - дизайн приложения;
 - описание классов, созданных для реализации приложения;
 - использование методов классов в приложении, взаимодействие экранных компонентов с объектами классов;
 - описание работы с приложением.
- **Заключение** содержит краткие выводы по результатам выполненной работы, оценку полноты решения поставленных задач, рекомендации по использованию приложения, целевую аудиторию.

Автономная некоммерческая организация дополнительного
профессионального образования
«Открытый молодёжный университет»

**КОМПЛЕКСНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
«ШКОЛЬНЫЙ УНИВЕРСИТЕТ»**

ПРОЕКТ

Разработка приложения калькулятора

Выполнили: учащиеся 10 «А» класса
МАОУ «Школа № 5»:
Игорь Сергеевич Васильев

Руководитель проектной деятельности:
преподаватель информатики
Пётр Михайлович Савинов

Томск — 2017

Техническое задание на проект

Проект: «Разработка приложения калькулятора».

Дата выдачи проекта: _____.

Разработчик проекта — Игорь Сергеевич Васильев.

Описание проекта

Описание проекта, согласно выбранному либо собственному заданию...

Задачи

1. Разработать интерфейс.
2. Разработать модель классов, которые описывают сущности проекта.
3. Реализовать код классов и взаимодействие их с экранными компонентами приложения.

Сроки сдачи проекта

1. Сбор материала (дата).
2. Распределение задач (дата).
3. Разработка интерфейса приложения (дата).
4. Выделение сущностей, которые могут быть описаны в виде классов (дата).
5. Разработка классов приложения (дата).
6. Разработка конечного приложения и его отладка (дата).
7. Подготовка пояснительной записки (дата).
8. Защита проекта (дата).