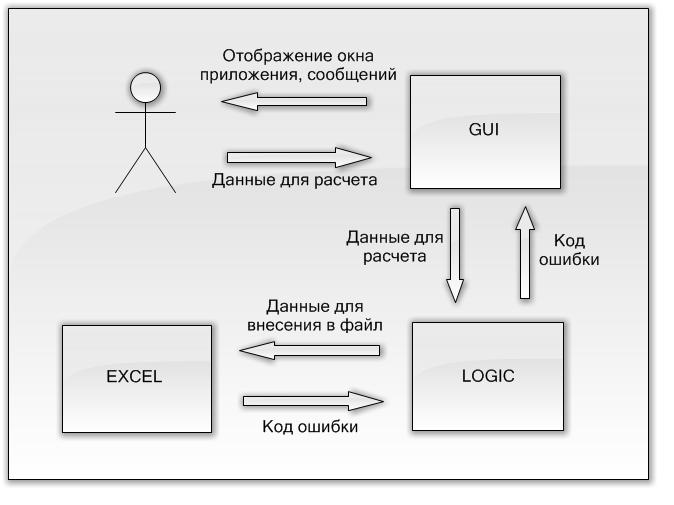
Приложение «Estimates 0.1.0»

Описание общей структуры

Приложение состоит из трех основных модулей:

1. GUI: модуль, отвечающий за отображение графического интерфейса приложения, за реакцию на действия пользователя, за проверку вводимых пользователем данных.
2. LOGIC: модуль, реализующий логику расчета смет из введенных пользователем данных.
3. EXCEL: модуль, ответственный за создание итогового файла формата xls.

Взаимодействие между модулями и пользователем показано на схеме:

Приложение пишется на языке Java. Исходный код приложения разбит на четыре корневых пакета:

1. *guipackage* — содержит реализацию модуля GUI.
2. *logicpackage* — содержит реализацию модуля LOGIC.
3. *excelpackage* — содержит реализацию модуля EXCEL.
4. *supportpackage* — вспомогательный пакет.

Общие положения

Имена пакетов, подпакетов, классов, методов, полей пишутся на английском языке.

Имена методов являются глаголами, отражающими суть задачи, выполняемой методом: *CalculateEstimates().*

Имена классов и полей являются существительными, отражающими моделируемую сущность либо роль в структуре приложения: *Material, ExcelHelper.*

Имена пакетов и подпакетов пишутся строчными буквами.

Имена классов, методов, полей пишутся с заглавной буквы. Если имя содержит несколько слов, каждое из них пишется с заглавной буквы.

Методы (кроме конструкторов и методов обратного вызова) и поля классов объявляются приватными, если в задании не указано обратное.

Задание на разработку модуля LOGIC.

Модуль реализует логику расчета смет из введенных пользователем данных.

В пакете *logicpackage* должен содержаться класс *public class LogicHelper*, имеющий конструктор вида *LogicHelper(String Path, String Filename, int EmptyLines, double Measure)* . Значения *Path, Filename, EmptyLines* не используются в модуле, а только передаются модулю EXCEL. *Measure* означает площадь потолка, подлежащего обсчету, измеряемую в квадратных метрах.

Класс *LogicHelper* должен содержать метод *public* *int CalculateEstimates(),* вызываемый извне модуля. Метод рассчитывает количество материалов, необходимых для отделки потолка, площадь которого получил конструктор.

Расчет ведется по следующему алгоритму:

На 3,6 кв.м. потолка необходим один лист ГКЛ. Число необходимых листов ГКЛ округляется в большую сторону. На один лист ГКЛ необходимо:

1. профили направляющие 60/27: 4 шт.
2. саморезы 3.2 x 25 мет.: 75 шт.
3. соединительные профили: 3 шт.
4. подвесы: 18 шт.
5. саморезы 4.2 x 16: 93 шт.

Результатом работы метода является массив объектов класса *supportpackage.Material.* Для каждого из необходимых материалов (в том числе и для листов ГКЛ) должен быть создан один объект.

public class Material {

public String Name;

public String Unit;

public int Amount;

public double Price;

Material() {

Name = "";

Unit = "";

Amount = 0;

Price = 0;

}

Material(String Name, String Unit, int Amount, double Price) {

this.Name = Name;

this.Unit = Unit;

this.Amount = Amount;

this.Price = Price;

}

}

При этом:

1. Name — имя материала;
2. Unit — единица измерения материала;
3. Amount — суммарное необходимое количество материала;
4. Price — цена материала в рублях.

Для версии 0.1.0 цены выбираются условными, из диапазона 1-10000.

После формирования массива, создается объект класса *excelpackage.ExcelHelper .* Конструктор класса имеет следующий вид: *ExcelHelper(String Path, String Filename, int EmptyLines, double Measure, Material[] Materials).* Значения *Path, Filename, EmptyLines, Measure* передаются конструктору так, как они были получены модулем. В качестве *Materials* передается сформированный массив объектов *Material.*

После создания объекта класса *ExcelHelper,* вызывается метод *public* *int CreateEstimates()* данного объекта*.* Возвращаемое методом значение является кодом ошибки.

Метод *CalculateEstimates()* должен вернуть значение:

1. 0, если в ходе работы модуля не произошло ошибок, и функция *CreateEstimates()* вернула значение 0.
2. 1, если в ходе работы модуля произошла ошибка, либо функция *CreateEstimates()* вернула значение, отличное от нуля.

Допустимо создание других классов или подпакетов при необходимости.