Федеральное агентство по образованию

Санкт-Петербургский государственный политехнический университет

—

Факультет технической кибернетики

**Кафедра «Компьютерных систем и программных технологий»**

**ОТЧЁТ**

**по курсовому проекту "Создание модели солнечной системы"**

**на тему «Программирование на языке Java»**

Выполнил

студент гр.2081/3 Лысенко К.В.

Руководитель

доцент, к.т.н. Пышкин Е. В.

Санкт-Петербург

2012

**Техническое задание:**  
Программа предназначена для симуляции солнечной системы; в ней планеты должны изменять свою скорость по желанию пользователя ;Периоды обращения планет должны задаваться в годах; Также в ней должны быть предоставлены данные о каждой планете; Программа должна использовать 3 формата файлов: файл, представляющий собой изображение планеты в формате .gif, файл, содержащий описание планеты, в формате .txt и файл, содержащий параметры планет в формате .dat; Интерфейс должен представлять собой окно с вращающимися объектами так же пользователю должна быть предоставлена возможность изменять размеры окон. В приложении должны быть реализованы классы для представления планет, траекторий, по которым они движутся; Планета характеризуется размером, названием, изображением, описанием. В программе должен быть реализован вид траектории движения по окружности, который характеризуется радиусом окружности, местоположением центра окружности, скоростью.

**Документ о границах проекта:**

**Образ и границы проекта**

Программа должна представлять собой модель солнечной системы. Планеты должны вращаться вокруг Солнца с разной скоростью и по разным орбитам. Планета представляет собой движущееся изображение данной планеты. Планеты должны быть "кликабельны" и при клике мышью на планету должно открываться окно с увеличенным изображением этой планеты и кратким описанием. Должна быть предусмотрена возможность изменения скорости. Изображение должно масштабироваться при изменении окна. Параметры планеты (такие как скорость, радиус , период обращения т.д) должны задаваться в файле.

**Структура исходных данных:**  
Исходные данные должны представлять собой систему из папок и файлов; Должны быть предусмотрены файлы, содержащие данные о координатах и параметрах движения планеты; файлы, содержащие описание планет; файлы, представляющие собой изображения планет в увеличенном и уменьшенном видах; Названия папок должны соответствовать названию планет.

**Основные сценарии функционирования программы(конкретизация сценариев):**   
1)Клик мыши(вызывается обработчик MouseClicked; при этом затрагиваются данные всех планет для того, чтобы определить находиться ли точка, в которую мы кликнули, в пределах какой-либо планеты и, если это произошло, создается это окно, в котором демонстрируются данные о планете(при закрытии этого окна с данными окно удаляется с экрана и очищается память))  
2)Использование стрелок "вверх" и,"вниз" и "пробел"(при этом затрагивается множитель Tfact ; при нажатии стрелки "вверх" скорость вращения планет увеличивается в два раза; при нажатии стрелки "вниз" скорость уменьшается в два раза; при нажатии знака "пробел" скорость вращения становиться обратно стандартной )  
3)Изменение размера окон(при этом затрагиваются данные Sfact; Каждый раз при перерисовке окна размеры окна сравниваются с базовым размером и, если есть отличие базового от действительного, то множитель Sfact пропорционально меняется)

**Основные классы приложения:**

Должен быть класс, представляющий главное окно приложения: в нем должны содержаться все планеты; Также должен быть класс представляющий объект, который будет двигаться по экрану; Должен быть класс, в котором содержаться данные о траектории движения планет; Должен присутствовать класс, представляющий окно с информацией об определенной планете.

**Интерфейс пользователя:**

Представляет собой окно с вращающимися объектами;

**Системная архитектура**

**1)Graphic Pack-содержит классы, представляющие графический интерфейс**

**1.1)BaseScreen  
1.2)MainFrame  
1.3)MainPanel  
1.5)PlanetFrame  
1.6)PlanetPanel  
1.7)Screen**

**1.8)KeyMonitor**

**1.9)TimerListener**

**1.10)MLIstener**

**1.11)FrameCloser**

**2) Logic Pack-содержит классы, реализующие внутреннюю логику приложения**

**2.1)Planet  
2.2)SpaceObject  
2.3)TrajCircle  
2.4)TrajStat  
2.5)Trajectory**

**3)sunsustem-содержит главный класс приложения и класс, организующий загрузку данных.**

**3.1)PlanetLoader  
3.2)SunSystem**

**3.3)Filter  
3.4)NoPlanetException**

*1)MainFrame – класс представления главного окна;*

*2)KeyMonitor– обработчик событий нажатия клавиш клавиатуры;*

**Содержание класса MainFrame:**

Содержит в себе панель, в нем находится обработчик события нажатия клавиатуры

**Содержание класса KeyMonitor:**

При нажатии определенных клавиш на клавиатуре данный класс изменяет временной множитель, который определяет скорость вращения относительно базовой, умножая его на два или деля на два, чтобы ускорить или замедлить скорость движения планет.

*3)PlanetFrame –рамка окна в которой выводиться информация о планете;*

*4)FrameCloser–обработчик события закрытия окна ;*

**Cодержание класса PlanetFrame:**  
содержит в себе панель, на которую выводиться информация о планете: увеличенное изображение планеты и описание планеты.

**Cодержание класса PlanetCloser:**  
при нажатии кнопки закрытия окна он удаляет окно с экрана и освобождает память.

*5)MainPanel – содержит коллекцию планет, временной и экранный множители, шаг и параметры BaseSize на основании которого будет отслеживаться изменение размера окна; выводит на экран планеты.*

*6)TimerListener–таймер, который отсчитывает время между перерисовкой окна ;*

*7)MListener-обработчик событий мыши;*

**Содержание класса MainPanel**

Реализует интерфейс Screen предоставляя доступ к множителю(изменение, получение значения); Содержит обработчик события мыши, который реагирует на клик мышкой на экране. Содержит обработчик срабатывания таймера

**Содержание класса TimerListener**Реагирует на срабатывание таймера, вызывает метод для пересчета координат планет и перерисовки окна.

**Содержание класса MListener**

при нажатии мыши в окне, проверяет находиться ли координаты точки на которую мы кликнули в пределах какой-либо планеты.

*8)SpaceObject –представление графического объекта*

**Содержание класса SpaceObject**

содержит информацию о местоположении, траектории, радиусе, изображении некоего графического объекта. Он занимается тем что выводит самого себя на графический контекст и в соответствии с множителем, который определяет масштаб экрана(изменение размера экрана), он отвечает за проверку лежит ли интересующая точка в пределах этого круга.

*9)Planet -абстрактное представление планеты*

**Содержание класса Planet**

расширяет класс SpaceObject и добавляет имя планеты, ее увеличенное изображение и описание;

*10)PlanetLoader –загружает планеты из папки;*

*11)Filter-вспомогательный класс: нужен для нахождения в папке подпапок;*

*12)NoPlanetException-исключение, выбрасывающееся если папка не содержит файлов с информацией о планете.*

С**одержание класса PlanetLoader**

Пытается загрузить из папки с указанным названием планеты. Находит подпапки и загружает из них файлы, содержащие информацию о планете.

*13)BaseScreen –вспомогательный класс, который представляет собой экран по умолчанию.;*

*14)Screen-интерфейс, который объявляет методы для редактирования и получения доступа к параметрам экрана*.

**Содержание класса BaseScreen:**

При попытке доступа к множителям возвращает некоторые значения по умолчанию. Игнорирует попытки изменить их.

*15)SunSystem –главный класс приложения в котором создается главное окно;*

*16)Trajectory – это абстрактный класс от которого уже унаследованы различные виды траекторий;*

*17)TrajCircle-класс, реализующий один из видов траектории- движение по окружности; содержит в себе такие параметры, как цикл, направление, радиус и местоположение круга и Screen;*

*18)TrajStat-класс, реализующий один из видов траектории- статическая траектория;*

**Содержание класса TrajCircle**

Рассчитывает шаг исходя из временного множителя и временного шага и при вызове соответствующего метода высчитывает положение объекта в следующий момент времени.

*19)PlanetPanel – панель, на которую выводиться информация о планете(картинка и описание);*

**Содержание класса PlanetPanel:**

Выводит на экран картинку, информацию о планете.

**Описание форматов данных**

Вся информация о планетах храниться в файлах и загружается из них. Для загрузки каждой планеты используется четыре файла:

%НазваниеПланеты%– имя папки используется для названия планеты, так же в этой папке содержаться файлы "pData.dat"," descr.txt" и 2 изображения планеты "PlanetSmall.gif" и " PlanetBig.gif";

pData.dat – содержит информацию о планете (радиус планеты,статическая/динамическая траектория, направление движения, начальный угол, время обращения(в годах), координаты центра траектории вращения);

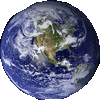
descr.txt – содержит описание планеты;  
PlanetBig.gif- картинка в описании планетыPlanetSmall.gif-картинка, которая представляет планету.

Пример:  
1)папка "Земля"

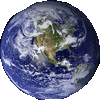
1.1)descr.txt

(Земля является крупнейшей и самой плотной из внутренних планет. У Земли наблюдается тектоника плит. Вопрос о наличии жизни где-либо, кроме Земли, остаётся открытым. Однако среди планет земной группы Земля является уникальной (прежде всего — гидросферой). Атмосфера Земли радикально отличается от атмосфер других планет — она содержит свободный кислород. У Земли есть один естественный спутник — Луна, единственный большой спутник планет земной группы Солнечной системы)

1.2)planetBig.gif



1.3)planetSmall.gif



1.4)pData.dat  
(15 0 152 true 0.00 1.0 400 400)

**Описание графического интерфейса:**

Графический интерфейс представляет собой окно с движущимися объектами и окна с изображениями планет и их описанием.

**Тестирование:**

1)проверка "кликабельности" планет:  
Запускаем программу; Для начала кликнем на пустое пространство окна: в результате ничего не происходит; далее кликнем на планету "Венера": в результате появляется окно с изображение планеты "Венера" и ее описанием.

Результат успешен.

2)проверка ускорения/замедления скорости движения планет.

Запускаем программу; Наблюдаем за движением планет при их базовой скорости; Нажимаем на клавишу "стрелка вверх": в результате скорость движения планет увеличивается в два раза; Нажимаем клавишу "пробел": в результате скорость движения планет становиться прежней(базовой); Нажимаем клавишу "стрелка вниз": в результате скорости движения планет уменьшается в два раза.  
Результат успешен.

3)проверка изменения размера окна.

Запускаем программу; Наводим стрелку мыши на границу окна и растягиваем ее: в результате окно увеличивается; Наводим стрелку на границу окна и сжимаем ее: в результате окно уменьшается. При обоих изменениях планеты сохраняют свои пропорции и размер относительно других планет.

Результат успешен.