Занятие 19 (2-й год)

Построение графиков функций

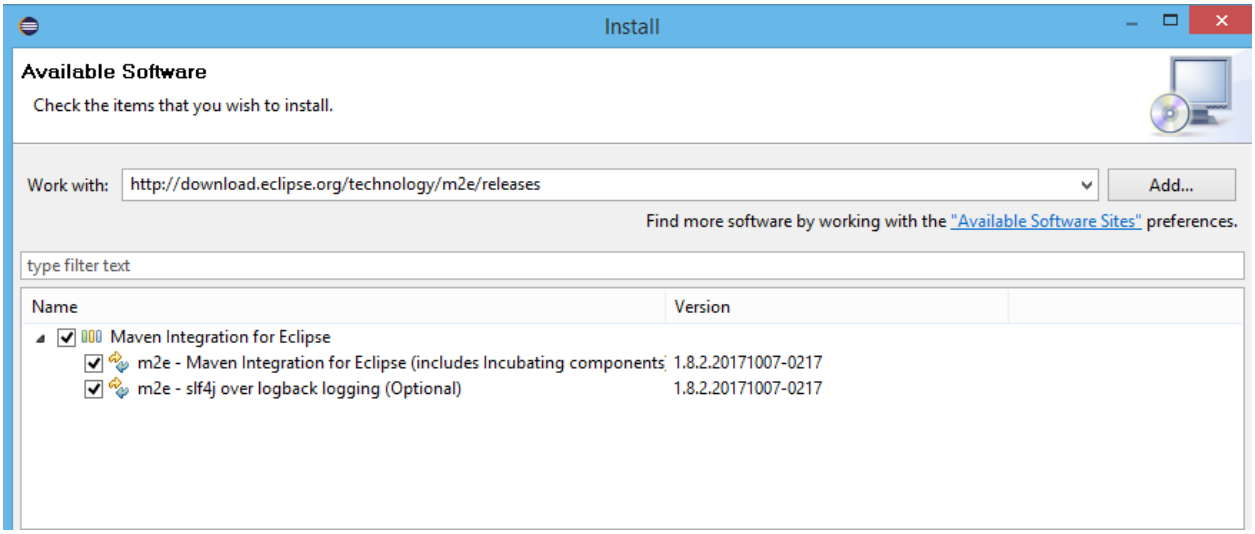
с помощью библиотеки JFreeChart

Хорошие новости: проект продолжает развиваться:

Скачать библиотеки, исходники, примеры и документацию новой версии:

https://github.com/jfree/jfreechart

http://download.eclipse.org/technology/m2e/releases



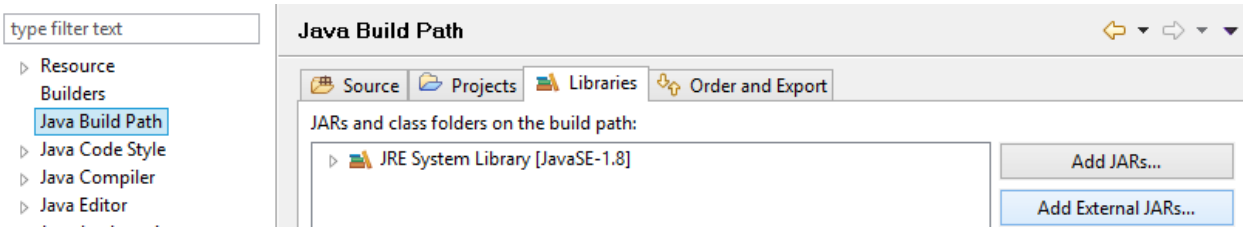
Старая версия:

https://sourceforge.net/projects/jfreechart/files/

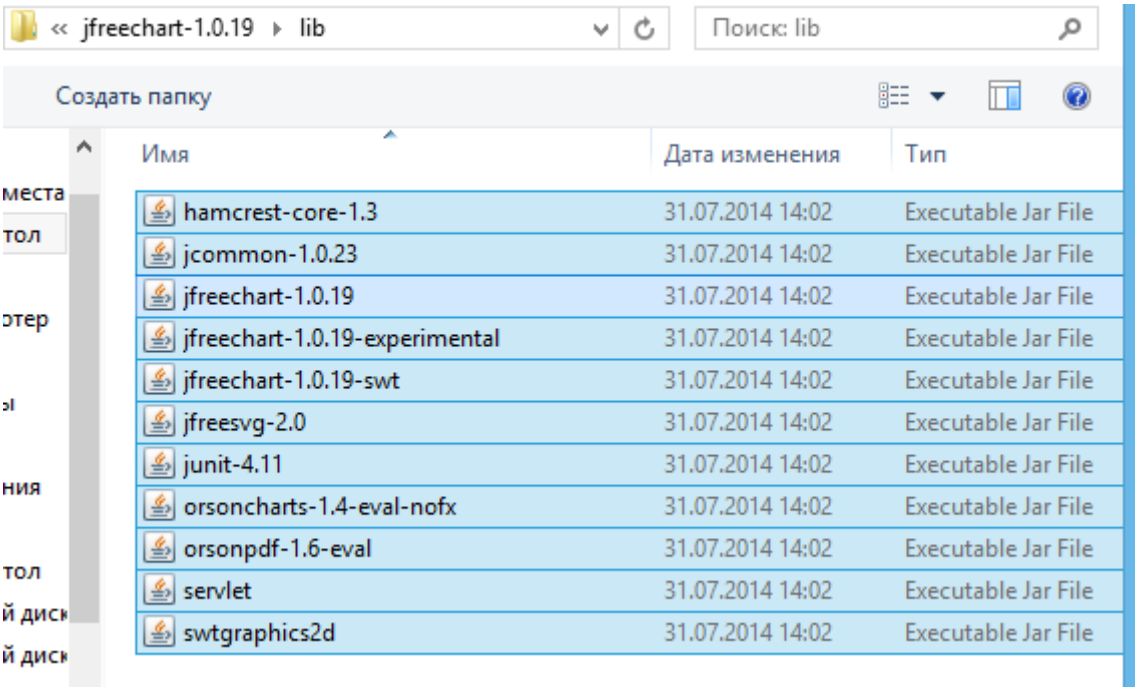
Разархивировать в удобном месте на компьютере.

Как подключить дополнительные библиотеки:

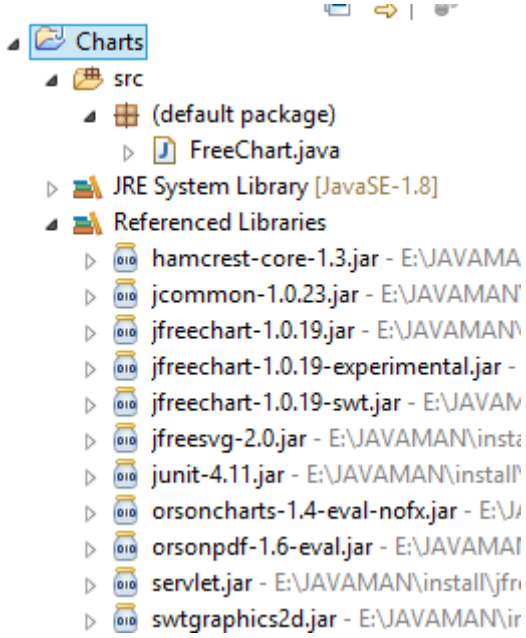
ПК по проекту – Properties



Add External JARS – выделить разархивированные библиотеки



Результат:



Появились дополнительные библиотеки

Пример 1 – Круговая диаграмма

Создадим класс, наследующий JFrame, для вывода круговой диаграммы.

1) Импорт классов из дополнительных библиотек:

```
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
```

2) Каждая созданная диаграмма-объект имеет тип JFreeChart.

```
//объект графика
JFreeChart chart1;
```

3) Для размещения графика служит специальная панель ChartPanel.

```
public void createChartPanel(){
    createPieChart();
    //создаем панель для графика
    ChartPanel cp=new ChartPanel(chart1);
    add(cp);
    pack();
    setTitle("Графики");
    setLocationRelativeTo(null);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setVisible(true);
}

public void createPieChart(){
    //исходные данные для графика
    DefaultPieDataset pd=new DefaultPieDataset();
    pd.setValue("A", 75);
    pd.setValue("B", 100);
    pd.setValue("C", 200);
    pd.setValue("D", 75);
    //строим график
    chart1=ChartFactory.createPieChart("Круговая диаграмма", pd, true, true, false);

    //сохранить график в изображение
    try {
        ChartUtilities.saveChartAsJPEG(new File("E:\\chart.jpg"), chart1, 500, 300);
        } catch (Exception e) {
        System.out.println("Problem occurred creating chart.");
        }
}
```

Практическая работа №1

С помощью WindowBuilder создать графический интерфейс для построения круговой диаграммы из 4-х сегментов:

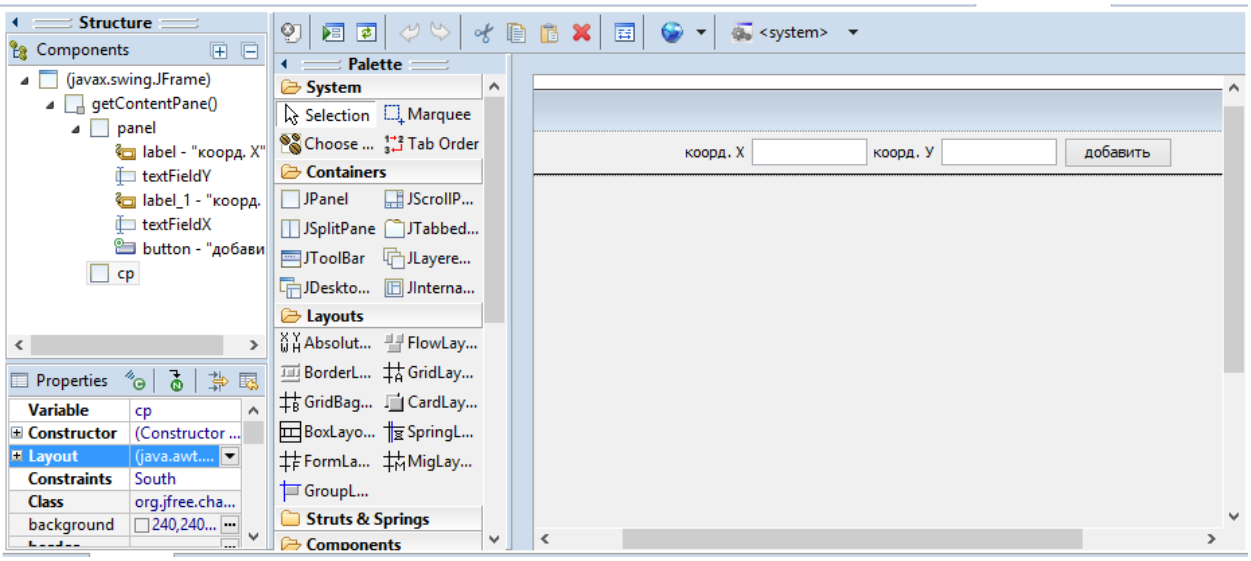
Примечание: двойной щелчок по кнопке добавляет к ней слушатель событий.

```
JButton btnNewButton = new JButton("\u041F\u043E\u0441\u0442\
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        a=Double.parseDouble(textField_3.getText());
        b=Double.parseDouble(textField_2.getText());
        c=Double.parseDouble(textField_1.getText());
        d=Double.parseDouble(textField.getText());
        createPieChart();
        cp.setChart(chart1);
    }
});
```

Практическая работа №2

Создадим приложение, в котором график будет строиться постепенно в результате текущих вычислений.

На верхней панели можно вводить данные, которые будут изображаться в виде графика на нижней ChartPanel-панели (по щелчку кнопки).



Шаг 1 – Создание ChartPanel-панели

```
cp = new ChartPanel(chart);
getContentPane().add(cp, BorderLayout.SOUTH);
```

Шаг 2 – Слушатель кнопки

```
public void actionPerformed(ActionEvent arg0) {
    createXYLine();
}
```

Шаг 3 – Метод построения линейного графика

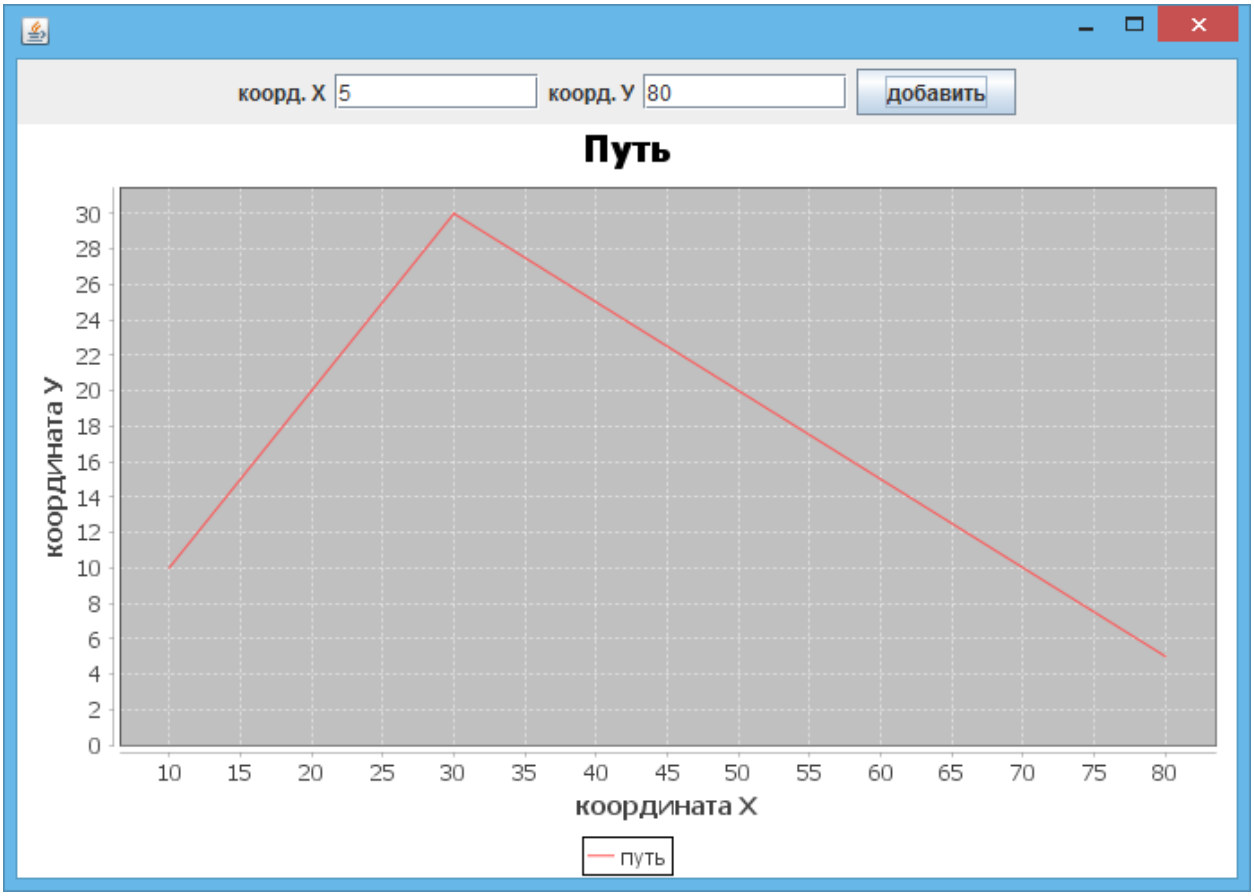
```
private void createXYLine() {

    //считать данные для графика
    double x=Double.parseDouble(textFieldX.getText());
    double y=Double.parseDouble(textFieldY.getText());
    //добавить данные в коллекцию
    ser.add(x, y);

    //добавить коллекцию в набор данных
    XYSeriesCollection dts = new XYSeriesCollection();
    dts.addSeries(ser);

    //построить график
    chart = ChartFactory.createXYLineChart("Путь", "координата X", "координата У", dts);
    cp.setChart(chart);
}
```

Результат:



Моделирование

Моде́ль (фр. *modèle*, от лат. *modulus* — «мера, аналог, образец») — это система, исследование которой служит средством для получения информации о другой системе[1]; представление некоторого реального процесса, устройства или концепции[2].

Модель есть абстрактное представление реальности в какой-либо форме (например, в математической, физической, символической, графической или дескриптивной), предназначенное для представления определенных аспектов этой реальности и позволяющее получить ответы на изучаемые вопросы[3]:80.

Термином моделирование обозначают как построение (создание) моделей, так и их исследование.

Одним и тем же системам могут быть сопоставлены несколько моделей разных видов.

Моделирование биологических процессов: биоритмы человека

1. **Постановка задачи**

1.1 Описание задачи

Существует гипотеза, что жизнь человека подчиняется трем циклическим процессам, называемым биоритмами. Эти циклы описывают 3 стороны самочувствия человека: физическую, эмоциональную и интеллектуальную. Биоритмы характеизуют подъемы и спады нашего состояния (синусоидальная зависимость). Дни, в которые график проходит через ось абсцисс, считаются неблагоприятными.

За точку отсчета всех трех биоритмов берется день рождения человека. Момент рождения для человека очень труден, ведь все три биоритма в тот день пересекают ось абсцисс.
Физический биоритм характеризует жизненные силы человека, его физическое самочувствие. Периодичность его составляет 23 дня.
Эмоциональный биоритм характеризует внутренний настрой человека.
Продолжительность эмоционального цикла равна 28 дням.
Интеллектуальный биоритм характеризует мыслительные способности. Цикличность его – 33 дня.
Предлагается осуществить моделирование биоритмов для конкретного человека от указанной текущей даты (дня отсчета) на месяц вперед с целью дальнейшего анализа модели.

1.2 Цель моделирования

На основе анализа индивидуальных биоритмов прогнозировать неблагоприятные дни, выбирать благоприятные дни для различной деятельности.

1.3 Формализация задачи

Объектом моделирования может быть любой человек, дата рождения которого известна.
Контрольные вопросы:
- что моделируется? – процесс изменения состояния человека.
- чем характеризуется челоек? – датой рождения.
- какое состояние исследуется? – физическое, эмоциональное, интеллектуальное.
- что известно о характере изменения состояний? – синусоидальное изменение с периодом 23, 28 и 33 дня.
- с каким шагом исследуется синусоида? – 1 день.
- какой период жизни исследуется? – 30 дней, начиная с текущего дня.
- что надо определить? – дни, когда кривые пересекают ось абсцисс и значения «взлетов».

2. Разработка модели

Информационная модель:

Исходные данные	Расчетные данные	Результаты
День рождения, день отсчета, длительность прогноза	Количество прожитых дней(x)	Физический, эмоциональный, интеллектуальный биоритм

Результаты описываются следующими выражениями:
$F(x)=\sin(2\pi x/23)$;
$E(x)=\sin(2\pi x/28)$;
$I(x)=\sin(2\pi x/33)$;

Компьютерная модель

Для моделирования разработана программа на Java. Этапы моделирования:

1. Строим графики функций биоритмов, используя метод `createXYLineChart()` на основе `XYSeriesCollection` без привязки к дате рождения.

```
public class BioR extends JFrame{
    JFreeChart chartF;
    ChartPanel cp;

    public static void main(String[] args) {
        new BioR();
    }
    BioR(){
        createChartPanel();
    }
public void createChartPanel(){

public void createSinChart(){
    //исходные данные для графика
    final XYSeries seriesF = new XYSeries("Физические биоритмы");
    for(double i = 0; i <=30; i ++){
        double fB = Math.sin(2*Math.PI*i/23);
        seriesF.add(i, fB);
        System.out.println(fB);
    }
    //коллекция на основе исходных данных
    final XYSeriesCollection datasetF = new XYSeriesCollection();
    datasetF.addSeries(seriesF);

    //строим график на основе коллекции данных
    chartF=ChartFactory.createXYLineChart("Биоритмы", "день", "биоритм",datasetF);
}
```

2. Вычисляем количество прожитых дней на данный момент.

java.time.LocalDateTime

Класс java.time.LocalDateTime хранит дату и время. Он является нечтом вроде комбинации LocalDate и LocalTime. В дополнение к методу now(), который есть у каждого временного класса, класс LocalDateTime содержит большое количество методов of, которые позволяют создать экземпляры LocalDateTime. Метод from конвертирует экземпляр другого класса в LocalDateTime. Также есть методы для добавления и вычитания часов, минут, дней и недель

3. Пересчитываем графики.
4. Создаем графический интерфейс в WindowBuilder для ввода даты рождения и построения графиков.

3. Компьютерный эксперимент
   План эксперимента
   - тестирование: по результатам расчетов построить общую диаграмму для трех биоритмов.
   - эксперимент: провести расчеты для разных исходных данных. Исследовать показания биоритмов.

4. Анализ результатов моделирования
   - составить прогноз благоприятных и неблагоприятных дней для различных видов деятельности (спорт, общение с друзьями, научная работа).
   - определить совместимость людей по биоритмам (близость графиков)