Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по дисциплине:
Методы математического моделирования
сложных процессов и систем

Студент	Абидоков Рашид Ширамбиевич
Группа	РК6-11М
Тип задания	лабораторная работа
Тема лабораторной работы	Разработка распределенных расширяемых программных систем

Студент _____ **Абидоков Р. Ш.**
подпись, дата *фамилия, и.о.*

Преподаватель _____ **Соколов А. П.**
подпись, дата *фамилия, и.о.*

Оценка _____

Москва, 2020 г.

Оглавление

Задание на лабораторную работу

Доработать реализованное ПО, созданное в рамках лабораторной работы №1 так, чтобы функции, обеспечивающие решение ОДУ подключались автоматически, а при их отсутствии не происходило аварийного завершения работы программы. Аналогичное требование к функциям, обеспечивающим вывод результатов расчетов. Среди функций, реализующих вывод результатов должны быть созданы:

- функция вывода результата в консоль
- функция вывода результата в виде .csv файла
- выгрузку результата расчета на удалённый узел по протоколу FTP (настройки подключения не должны быть зашиты в код программы) (плагин-«визуализатор») (дополнительный модуль расширения, разрабатывать по желанию)

Цель выполнения лабораторной работы

Цель выполнения лабораторной работы – усвоить принципы создания модулей расширения (плагинов, англ. plug-ins) функциональных возможностей программных систем.

Выполненные задачи

1. Обеспечена возможность подключения и вызова произвольного количества плагинов трех типов – "конфигураторы", "решатели", "виджеты".
2. Реализованы плагин-конфигуратор, взаимодействующий с config-файлом, плагины-решатели, реализующие методы Эйлера и Хойна, плагины-виджеты, выводящие результаты расчета в консоль и выходной .csv-файл.
3. Дополнительно реализован плагин-виджет, осуществляющий загрузку файла решения на удаленный сервер с помощью FTP.

4. Обеспечена работоспособность программы при отсутствии плагинов определенных типов, а также при ошибках, связанных с их инициализацией.

1. Общее описание программной реализации

Программная реализация состоит из основного модуля main.exe и шести файлов с расширением .plg, являющихся плагинами. Модуль main.exe осуществляет поиск в заданной папке всех файлов с расширением .plg, регистрацию плагинов (т.е. поиск и вызов функций регистрации с последующим определением типов плагинов и распределением их по соответствующим контейнерам), а затем последовательный вызов одного из конфигураторов, всех решателей и всех виджетов. Иерархия реализованных плагинов приведена на Рис. 1, интерфейс базового плагина – в Листинге 1.

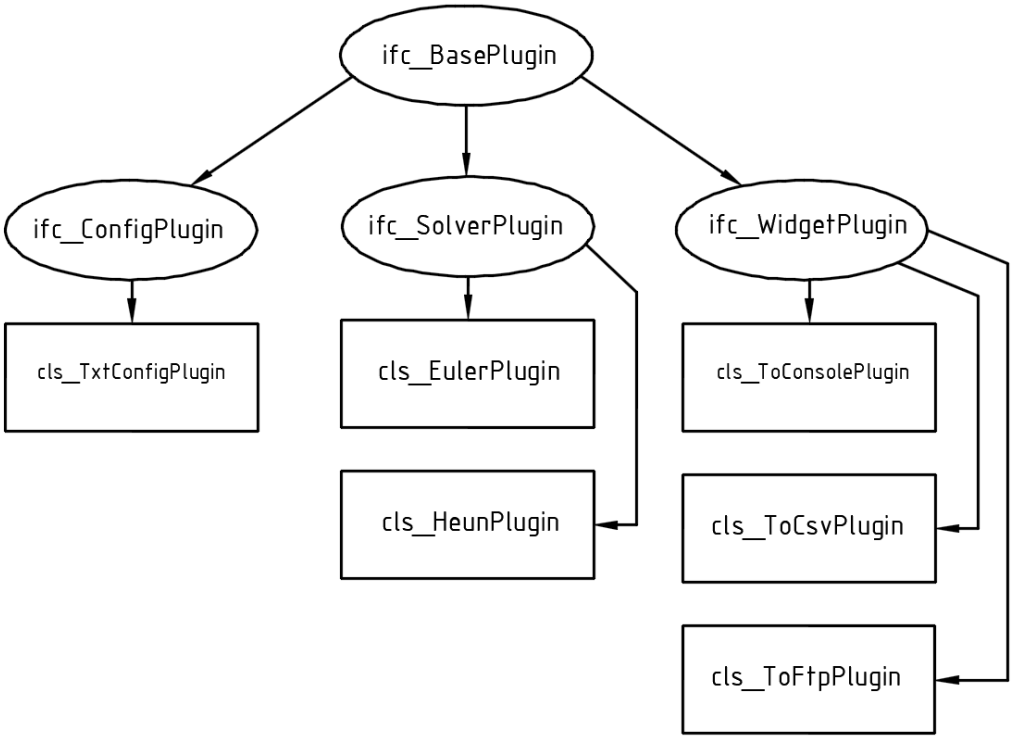


Рис 1.

Листинг 1.

```
class ifc_BasePlugin {
public:
  virtual int get_id() const = 0;
  virtual const char* get_name() const = 0;
  virtual const char* get_type() const = 0;
  virtual const char* get_title() const = 0;
  virtual bool is_instance(const char* req_name) const = 0;
  virtual ~ifc_BasePlugin() {};
};
```

В случае, если нет ни одного плагина определенного типа, происходит соответствующая обработка – при отсутствии конфигураторов используются параметры по умолчанию, при отсутствии решателей – рассчитывается только аналитическое решение, при отсутствии виджетов – решение не выводится (т.к. вывод результата в консоль также реализован отдельным плагином).

После выполнения программы в консоль печатается краткий отчет, содержащий в себе параметры модели и зарегистрированные плагины. Пример вывода отчета приведен на Рис. 2.

```
----------------------------
REPORT

Model parameters:
        gam_type: 1
        gam: 0.1
        T_s: 25
        T_beg: 150
        time_beg: 0
        time_end: 100
        n_steps: 1000

Console output:
        True (see above)

File output settings:
        out_file_name: out.csv
        csv_dlm: ,

FTP output settings:
        host: 127.0.0.1
        port: 21
        login: admin
        password: 12345678
        file_name: out.csv

Registered plugins:
ifc_ConfigPlugin
        cls_TxtConfig - txt config reader
  used: cls_TxtConfig
ifc_SolverPlugin
        cls_EulerSolver - Euler method
        cls_HeunSolver - Heun method
ifc_WidgetPlugin
        cls_ToConsolePlugin - prints the result to the console
        cls_ToCsvPlugin - .csv writer
        cls_ToFtpPlugin - ftp client
----------------------------
```

Рис 1.

2. Плагины-конфигураторы

Плагины-конфигураторы обеспечивают задание параметров математической модели, а также параметров, необходимых некоторым плагинам решателям. Интерфейс плагина-решателя приведен в Листинге 2. Непосредственный запуск плагина осуществляется вызовом функции config, возвращающей структуру, полями которой являются заданные параметры.

Листинг 2.

```
class ifc_ConfigPlugin : public ifc_BasePlugin {
public:
  virtual int get_id() const = 0;
  virtual const char* get_name() const = 0;
  virtual const char* get_type() const = 0;
  virtual bool is_instance(const char* req_name) const = 0;
  virtual ~ifc_ConfigPlugin() {};
  // Функция задания глобальных параметров
  virtual config_struct config(const char* settings = "") = 0;
};
```

Реализованный конфигуратор cls_TxtConfigPlugin обеспечивает чтение текстового файла config.txt и установку параметров. При отсутствии такого файла создает файл с настройками по умолчанию, структура которого приведена в Листинге 3.

Листинг 3.

```
        # Тип задания коэффициента гамма математической модели
        # 1 для точного значения, 2 для интервала, 3 для M(gam) и D(gam)
gam_type=1
        # Коэффициент гамма математической модели. Для gam_type=1 - одно число, для 2 и 3 - два числа через запятую
gam=0.1
        # Температура окружающей среды
T_s=25.
        # Начальная температура тела
T_beg=150.
        # Начальное время
time_beg=0.
        # Конечное время
time_end=100.
        # Желаемое количество шагов интегрирования
n_steps=1000
        # Настройки для виджета, выводящего результаты в файл
        # Имя выходного .csv файла
out_file_name=out.csv
        # Разделитель в выходном файле
csv_dlm=,
        # Настройки для виджета, передающего результаты по FTP
        # Имя создаваемого на сервере .csv файла
ftp_file_name=out.csv
        # Хост
ftp_host=127.0.0.1
        # Порт
ftp_port=21
        # Логин
ftp_login=admin
        # Пароль
ftp_password=12345678
```

3. Плагины-решатели

Осуществляют численное решение. Работа обеспечивается посредством вызова функции solve(), возвращающей вектор пар время-температура. Интерфейс решателя приведен в Листинге 4.

Листинг 4.

```
class ifc_SolverPlugin : public ifc_BasePlugin {
public:
  virtual int get_id() const = 0;
  virtual const char* get_name() const = 0;
  virtual const char* get_type() const = 0;
  virtual bool is_instance(const char* req_name) const = 0;
  virtual ~ifc_SolverPlugin() {};
  virtual std::vector<std::pair<double, double> >& solve(double (*der_func)(double),
    double y0, double begin, double end, unsigned n_steps) = 0;
};
```

Реализованные плагины cls_EulerPlugin и cls_HeunPlugin представляют собой методы Эйлера и Хойна соответственно.

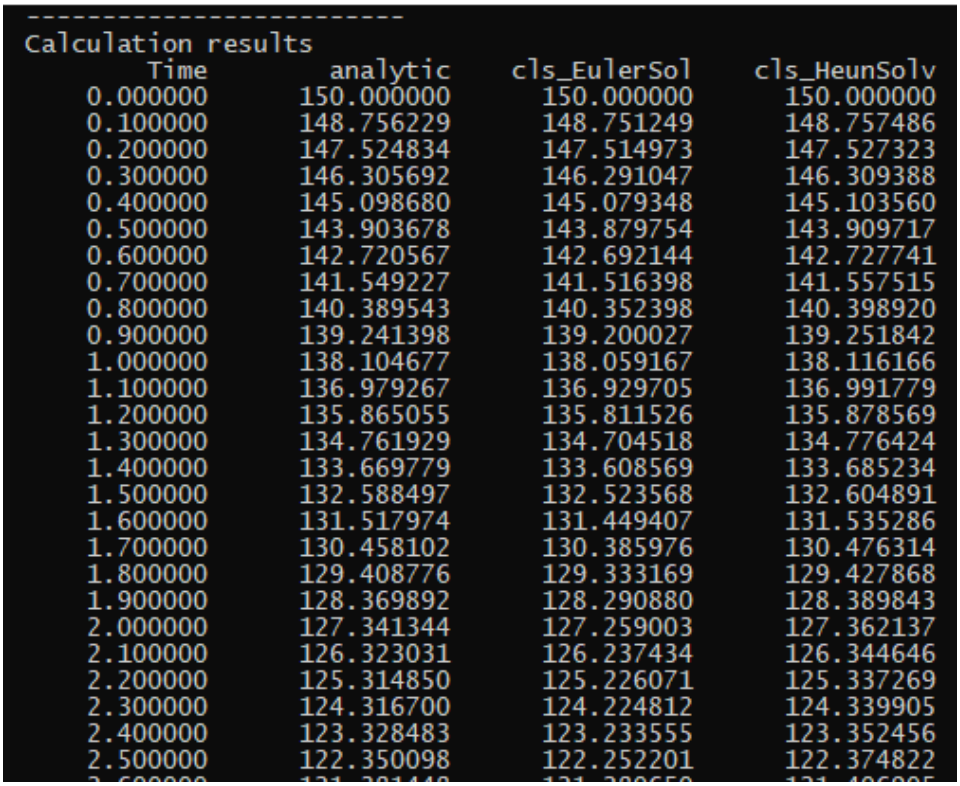
4. Плагины-виджеты

Осуществляют визуализацию результатов расчета. Интерфейс представлен в Листинге 5.

Листинг 5.

```
class ifc_WidgetPlugin : public ifc_BasePlugin {
public:
  virtual int get_id() const = 0;
  virtual const char* get_name() const = 0;
  virtual const char* get_type() const = 0;
  virtual bool is_instance(const char* req_name) const = 0;
  virtual ~ifc_WidgetPlugin() {};
  virtual int execute(map<string, vector<pair<double, double> > >& data, const char* settings = "") = 0;
};
```

Были реализованы три плагина-виджета. Вывод результатов в консоль осуществляет cls_ToConsolePlugin (пример вывода на Рис. 3). Запись в выходной .csv-файл обеспечивает cls_ToCsvPlugin. В свою очередь, cls_ToFtpPlugin производит загрузку результата на удаленный сервер с помощью FTP. Он написан с использованием библиотеки WinInet.



```
-------------------------
Calculation results
        Time        analytic    cls_EulerSol    cls_HeunSolv
    0.000000      150.000000      150.000000      150.000000
    0.100000      148.756229      148.751249      148.757486
    0.200000      147.524834      147.514973      147.527323
    0.300000      146.305692      146.291047      146.309388
    0.400000      145.098680      145.079348      145.103560
    0.500000      143.903678      143.879754      143.909717
    0.600000      142.720567      142.692144      142.727741
    0.700000      141.549227      141.516398      141.557515
    0.800000      140.389543      140.352398      140.398920
    0.900000      139.241398      139.200027      139.251842
    1.000000      138.104677      138.059167      138.116166
    1.100000      136.979267      136.929705      136.991779
    1.200000      135.865055      135.811526      135.878569
    1.300000      134.761929      134.704518      134.776424
    1.400000      133.669779      133.608569      133.685234
    1.500000      132.588497      132.523568      132.604891
    1.600000      131.517974      131.449407      131.535286
    1.700000      130.458102      130.385976      130.476314
    1.800000      129.408776      129.333169      129.427868
    1.900000      128.369892      128.290880      128.389843
    2.000000      127.341344      127.259003      127.362137
    2.100000      126.323031      126.237434      126.344646
    2.200000      125.314850      125.226071      125.337269
    2.300000      124.316700      124.224812      124.339905
    2.400000      123.328483      123.233555      123.352456
    2.500000      122.350098      122.252201      122.374822
```

Рис 3.

Заключение

Получен опыт реализации модулей расширения (плагинов). Реализована программа, осуществляющая численное решение ОДУ и обеспечивающая возможность расширения путем подключения произвольного количества плагинов.