

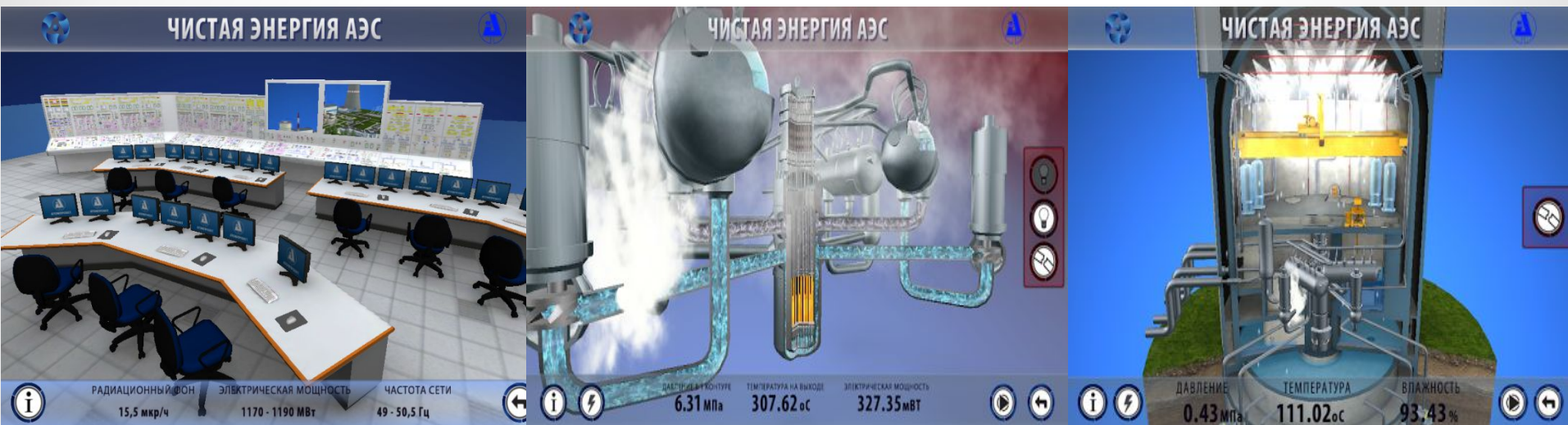
КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ РАБОТЫ ЯДЕРНОГО РЕАКТОРА

Авторы проекта : Ондасынов Арнат ,
Майлыбаева Мира

Школа : Областная
специализированная школа-интернат
для одаренных детей с углубленным
изучением различных предметов

Секция : Информатика

Цель исследования : Создание компьютерной программы, позволяющей моделировать изменение параметров работы ядерного реактора и управлять им.



ЭТАПЫ ИССЛЕДОВАНИЯ

- Проанализировать научную литературу по теме исследования;
- Изучить работу ядерного реактора;
- Создать компьютерную программу по моделированию работы ядерного реактора.

Гипотеза:

Если создать компьютерную программу, моделирующую параметры работы ядерного реактора, то появится возможность экспериментирования по изменению этих параметров в затрудненных финансовых и физических условиях, так как компьютерные модели проще и удобнее в использовании на производстве.

Новизна научной работы:

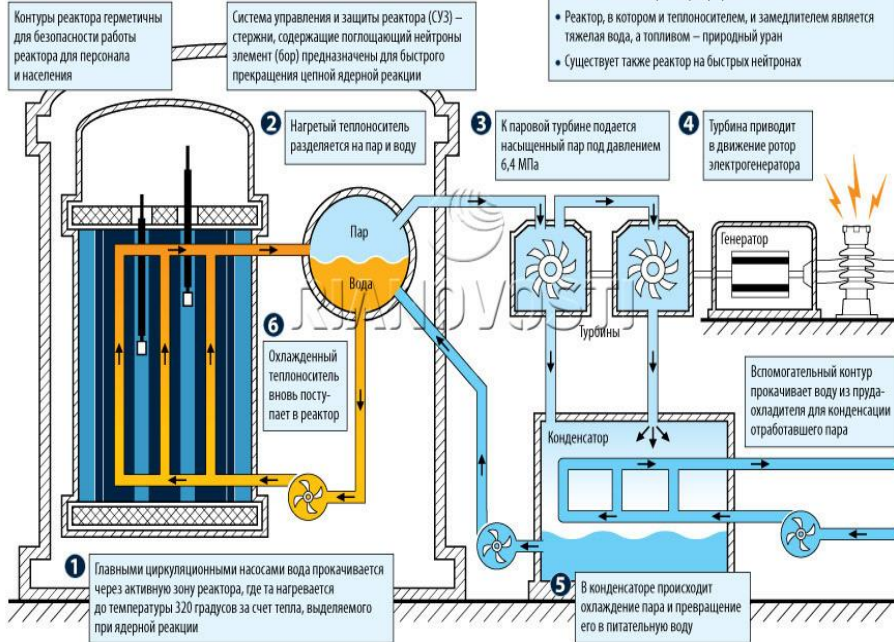
В настоящее время в Казахстане практически отсутствуют отечественные разработки в области управления ядерными реакторами.



Устройство атомной электростанции

Атомная электростанция (АЭС) – комплекс сооружений, предназначенных для выработки электрической энергии путем использования энергии, выделяемой при контролируемой ядерной реакции

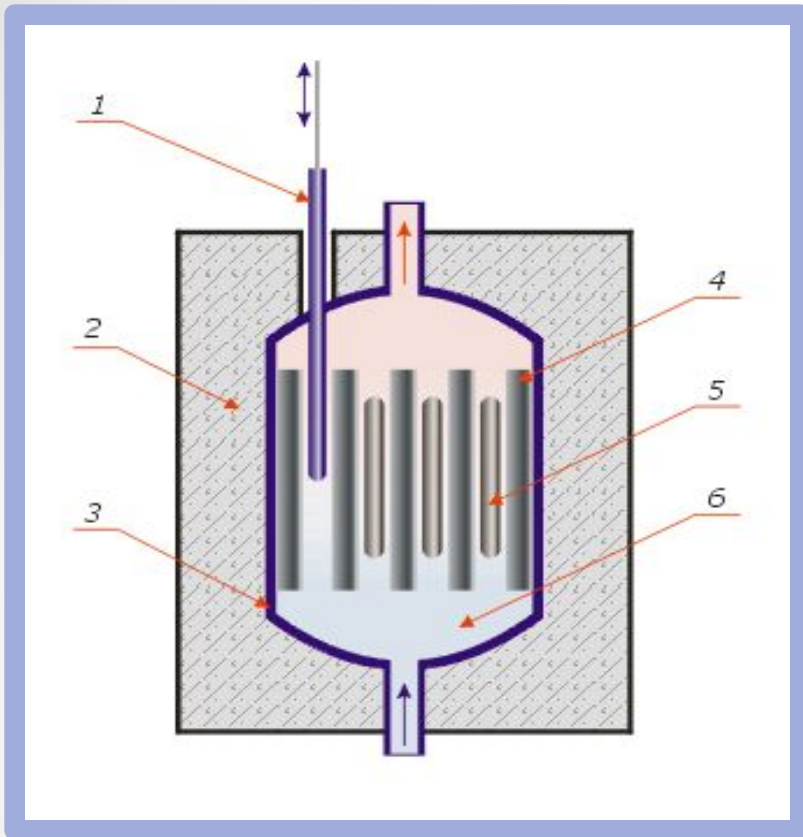
Основные процессы в работе АЭС



Основной блок АЭС – атомный реактор

- Легководный реактор:
 - кипятиль – пар, вращающий турбины, образуется в активной зоне (РБМК – реактор большой мощности, каналный)
 - водо-водяной – пар образуется во втором контуре, связанном с первым контуром теплообменниками и парогенераторами (энергетический реактор – ВВЭР)
 - Газоохлаждаемый реактор с графитовым замедлителем
- Реактор, в котором и теплоносителем, и замедлителем является тяжелая вода, а топливом – природный уран
- Существует также реактор на быстрых нейтронах

Атомная электростанция (АЭС) – ядерная установка для производства энергии в заданных режимах и условиях применения, располагающаяся в пределах определённой проектом территории, на которой для осуществления этой цели используются ядерный реактор (реакторы) и комплекс необходимых систем, устройств, оборудования и сооружений с необходимыми работниками (персоналом).



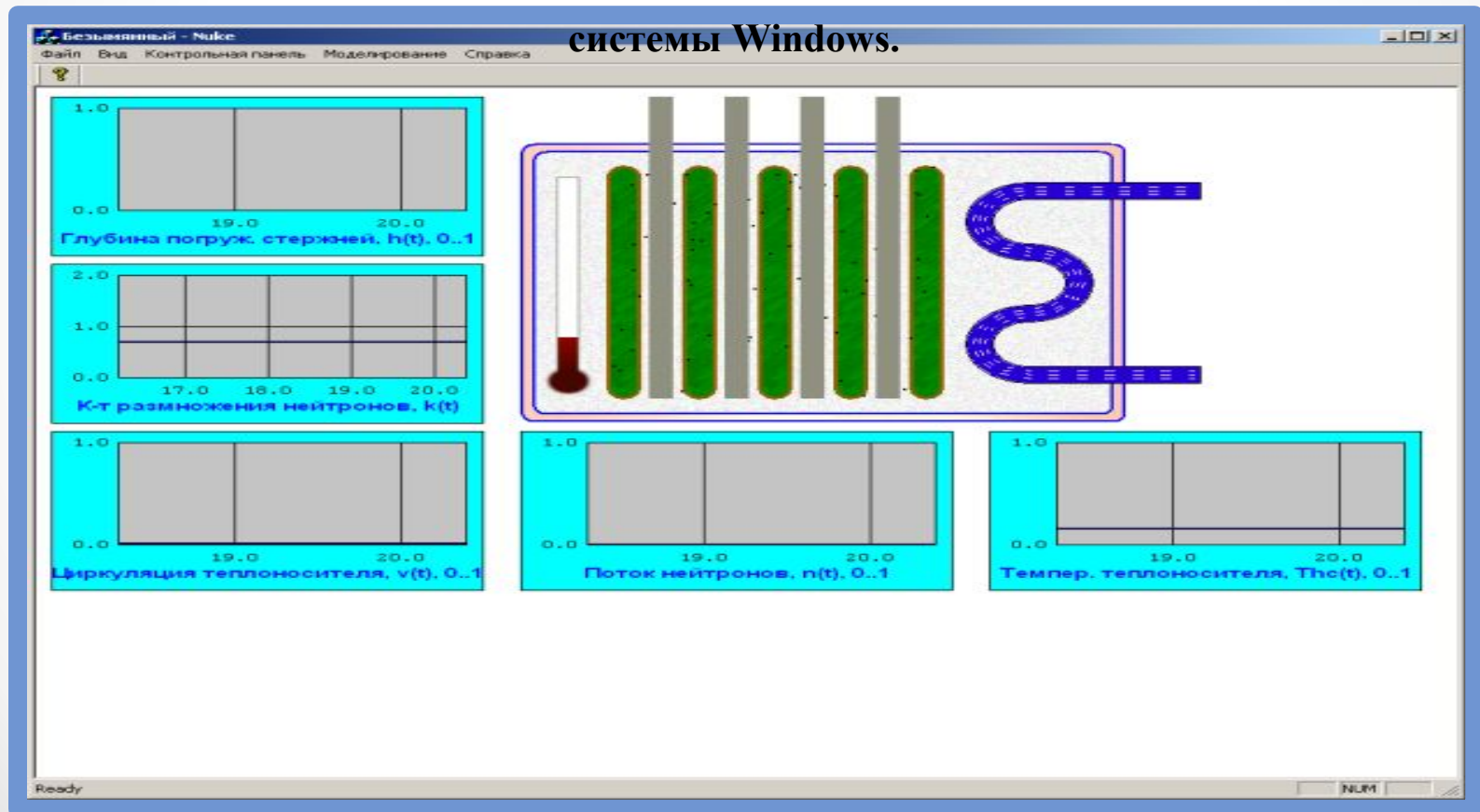
Ядерный реактор – это устройство, предназначенное для организации управляемой самоподдерживающейся цепной реакции деления, которая всегда сопровождается выделением энергии (1 МВт на $3 \cdot 10^{16}$ актов деления в секунду).

Схематическое устройство гетерогенного реактора на тепловых нейтронах
1-Управляющий стержень; 2- Радиационная защита; 3-Теплоизоляция; 4- Замедлитель; 5- Ядерное топливо; 6- Теплоноситель

Компьютерная модель включает в себя модель ядерного реактора с автоматическим управлением и контролем параметров реакции.

Разработка программы выполнена в среде разработки Microsoft Visual C++ 6.0.

В программе обеспечена возможность подачи команд оператором и моделирования нештатных ситуаций. Программа создана для работы под управлением операционной



Интерфейс пользователя

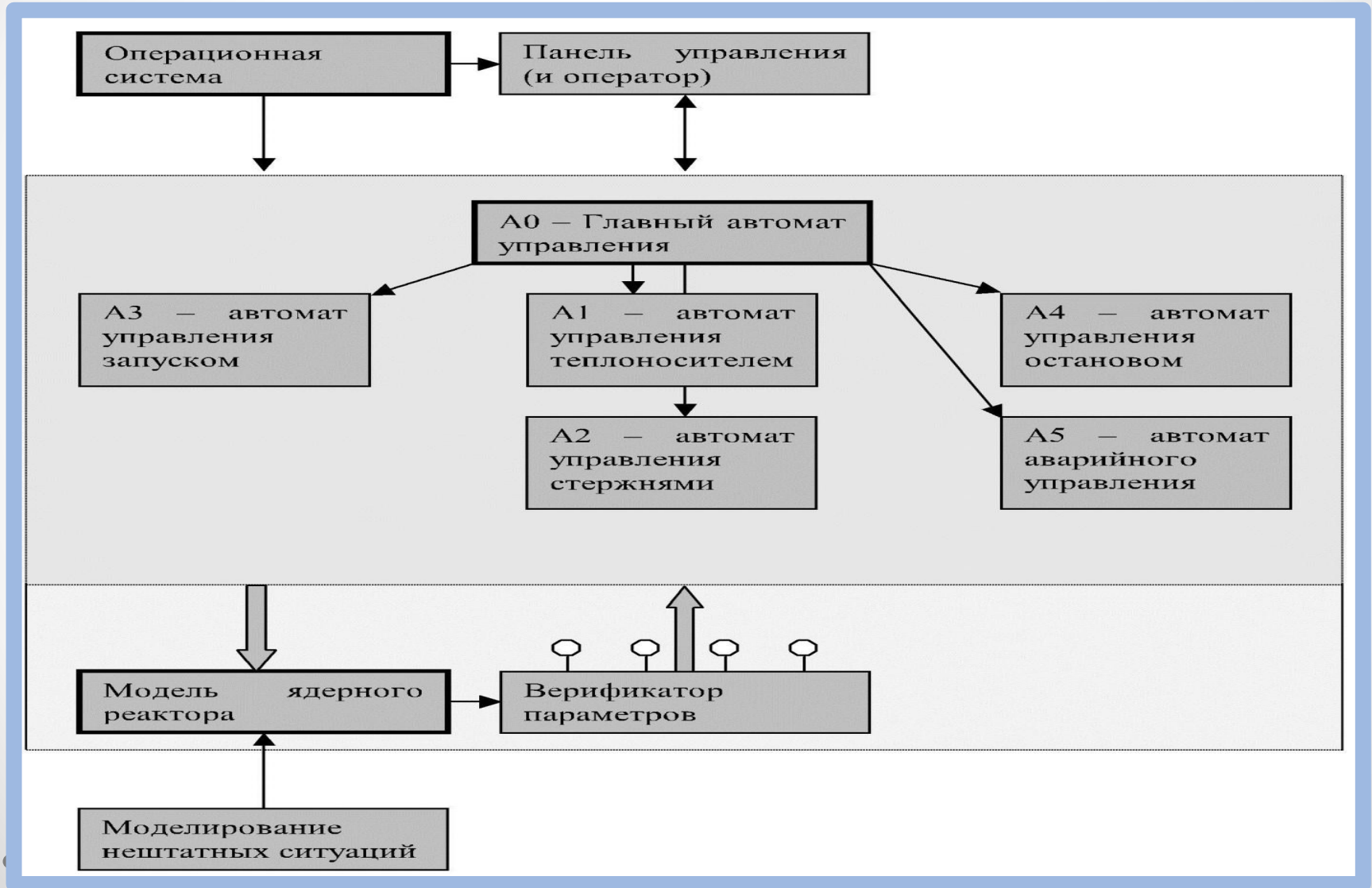
Функциональная спецификация системы

Система управления должна выполнять следующие функции :

- Оперативно реагировать на изменения условий протекания реакции путем выработки необходимых корректирующих воздействий ;
- Гарантировать поддержание всех важных параметров в допустимых пределах;
- При выходе параметров за пределы нормального диапазона попытаться восстановить нормальный режим работы;
- Если параметры работы реактора таковы, что ситуация не может быть исправлена, производить экстренную остановку реактора;
В критических ситуациях управления на себя должен брать блок защиты системы.

Описание логики управления

Логика управления сосредоточена в системонезависимой автоматной части.
Автоматы реализованы как отдельные функции.



Автомат управления теплоносителем (А1)

Как следует из названия, этот автомат управляет теплоносителем. Для этого он использует информацию о температуре и, частично, о количестве нейтронов. Логика этого автомата имеет схожую структуру с логикой автомата А2, управляющего стержнями. В качестве выходных воздействий в автомате используются функции «увеличить скорость теплоносителя» и «уменьшить скорость теплоносителя».

Автомат управления стержнями (А2)

Этот автомат управляет стержнями. Для этого он использует информацию о количестве нейтронов и о температуре. Логика этого автомата имеет схожую структуру с логикой автомата А1. В качестве выходных воздействий в автомате используются функции «увеличить глубину погружения стержней» и «уменьшить глубину погружения стержней».

Автомат управления запуском (А3)

Этот автомат вызывается из автомата А0, когда его состояние соответствует запуску реактора. Автомат А3 отвечает за действия, связанные с запуском реактора: он ничего не делает до тех пор, пока не будет произведена предпусковая инициализация ретых (не рассматриваемых в работе) систем (долговременные операции, такие как, разогрев труб). После этого автомат обеспечивает начальный разгон теплоносителя до определенной скорости с тем, чтобы автомат А0 мог перейти в состояние «Работа».

Автомат управления остановом (А4)

Этот автомат получает управление от автомата А0 в состоянии штатного останова реактора, которое происходит в случае нажатия оператором специальной кнопки на панели управления. Логика управления достаточно проста: сначала производится опускание стержней до максимума, затем реактор охлаждается (для этого теплоноситель разгоняется), а потом производится торможение теплоносителя.

Автомат аварийного управления остановом (А5)

Этот автомат, также как и автомат А4, управляет остановом, однако разница в том, что здесь останов экстренный. При первой передаче управления этому автомату включается аварийный звуковой сигнал. После этого автомат производит экстренные действия, связанные с быстрой нейтрализацией последствий факторов, вызвавших аварийную ситуацию.

Заключение

- В результате работы мы достигли своей цели и подтвердили гипотезу изложенные в абстракте.
- В виду недоступности некоторых технических характеристик ядерного реактора описанной в работе, модель получилась приближенная. Однако при наличии точных данных модель легко усовершенствовать. Результаты работы можно использовать в обучении школьников и студентов, связанных с решением задач атомной энергетики.
- Учитывая все вышесказанное, мы можем сделать следующие выводы:
- Компьютерное моделирование позволяет проводить эксперименты, реализация которых в реальности дорогостояще, длительно, труднодоступна.
- Полученные результаты позволяют с полной уверенностью утверждать, что в Казахстане возможна разработка программного обеспечения для управления ядерными реакторами.
- Поскольку большинство данных по атомной технике относится к государственной или военной тайне, компьютерная модель получилась приближенной.
- Данная работа может быть продолжена и улучшена при наличии точных данных и перерасти в серьезную исследовательскую работу, и служит для развития атомной энергетики Казахстана.

```

friend class Nuke_model; // модели должны быть доступны любые изменения параметров
private:
    timetype _time; // текущее время реактора (модели)
    // Сделаны графики
double _h; // глубина погружения стержней, в процентах (0-100)
    double _k; // коэффициент размножения (примерно равен 1)
    double _v; // скорость обращения теплоносителя, в процентах (0-100)
    double _n; // число вылетающих нейтронов, в процентах (0-100)
    double _Thc; // температура теплоносителя, в процентах (0-100)
    // Не сделаны графики
double _Twa; // температура рабочей зоны
    double _N; // тепловая мощность
double _P; // полезная мощность (электрическая)
    // смещения для основных параметров
double _dtime; // смещения по времени (вперед/назад)
    double _dh; // смещения глубины погружения (напр. стержень сломался)
    double _dk; // смещения к-та размножения (напр. дырка в реактора)
    double _dv; // смещения скорости теплонос (затор в трубах)
    double _dn; // изменения числа нейтронов (доп. источник)
    double _dTwa; // изменения темп. акт. Зоны (нарушен теплооток)
    double _dThc; // изменения темп. Теплоносителя (нарушен теплопоток)

public:
    // constructor
    inline Nuke_data ();
    // getters
    inline timetype time () const;
    inline double h () const;
    inline double k() const;
    inline double v() const;
    inline double n() const;
    inline double Thc() const;
    inline double Twa() const;
    inline double N() const;
    inline double P() const;
    // setters: будут реализованы лишь примитивные сеттеры, не требующие сложных расчетов. Все нетривиальные, а также зависимые от
    // выбранной модели расчеты будут производиться в Nuke_model
    //
    //
protected:

```

```

//-[ структура данных для блока управления ЯР
// доступны изменения:
// инкрементировать h – “погрузить стержни” на величину
// декрементировать h – “выдвинуть стержни” на величину
// установить скорость циркуляции теплоносителя
struct Csystem_data : public Nuke_data
{
public:
Csystem_data(const Nuke_data& data) : Nuke_data(data) {}

inline void cd_inc_h() { ibc_h(); }
inline void cd_dec_h() { dec_h(); }
inline void cd_set_v(double v) { set_v(v); }
inline void cd_inc_v() { inc_v(); }
inline void cd_dec_v() { dec_v(); }
};

//-[ структура данных для интерфейса пользователя
// доступны изменения;
// инкрементировать h – “погрузить стержни” на величину
// установить скорость циркуляции теплоносителя
struct Useriface_data : public Nuke_data
{
public:
Useriface_data(const Nuke_data& data) : Nuke_data(data) {}

inline void ud_inc_h() { ibc_h(); }
inline void ud_dec_h() { dec_h(); }
inline void ud_set_v(double v) { set_v(v); }

inline void ud_set_dtime(double dtime); { set_dtime(dtime); }
inline void ud_set_dh(double dh); {set_dh(dh);}
inline void ud_set_dk(double dk); {set_dk(dk);}
inline void ud_set_dv(double dv); {set_dv(dv);}
inline void ud_set_dn(double dn); {set_dn(dn);}
inline void ud_set_dTwa(double Twa); {set_dTwa(dTwa);}
inline void ud_set_dThc(double dThc); {set_dThc (dThc);}

```

```

// getters
inline timetype Nuke_data: : time() const {return _time + _dtime;}
inline double Nuke_data: : h() const { return _h + _dh; }
inline double Nuke_data: : k() const { return _k + _dk; }
inline double Nuke_data: : v() const { return _v + _dv; }
inline double Nuke_data: : n() const { return _n + _dn; }
inline double Nuke_data: : Thc() const { return _Thc + _dThc; }
inline double Nuke_data: : Twa() const { return _Twa + _dTwa; }
inline double Nuke_data: : N() const { return _N; }
inline double Nuke_data: : P() const { return _P; }

// some getters

inline void Nuke_data: :inc_time(timetype dt) ( _time += dt; )

inline void Nuke_data: :inc_h(){ _h = ( (_h<100) ? _h + PIVOT_H_STEP : 100); }
inline void Nuke_data: :dec_h() { _h = ( (_h>0) ? _h - PIVOT_H_STEP : 0); }

inline void Nuke_data: :set_v(double v) { _v = v; }
inline void Nuke_data: :inc_v(){ _v = ( (_v<100) ? _v + HEAT_CARRIER_V_STEP:100); }
inline void Nuke_data: :dec_v() { _v = ( (_v>1) ? _v - HEAT_CARRIER_V_STEP:1); }

// additional setters
inline void Nuke_data: :set_dtime(double dtime) { _dtime = dtime; }
inline void Nuke_data: :set_dh(double dh) { _dh = dh; }
inline void Nuke_data: :set_dk(double dk) { _dk = dk; }
inline void Nuke_data: :set_dv(double dv) { _dv = dv; }
inline void Nuke_data: :set_dn(double dn) { _dn = dn; }
inline void Nuke_data: :set_dTwa(double dTwa) { _dTwa = dTwa; }
inline void Nuke_data: :set_dThc(double dThc) { _dThc = dThc; }

inline void Nuke_data: :set_Twa(double Twa) { _Twa = Twa; }

// inline void Nuke_data: : set_p(double P) { _P = P; }

#endif // NUKE_DATA_H_

```