

Научная статья  
УДК 519.876.5  
DOI:[10.34759/trd-2022-123-22](https://doi.org/10.34759/trd-2022-123-22)

## **УНИФИЦИРОВАННАЯ ПРОГРАММНАЯ ПЛАТФОРМА ДЛЯ РАЗРАБОТКИ МНОГОАГЕНТНЫХ МОДЕЛЕЙ ОРБИТАЛЬНЫХ ГРУППИРОВОК КОСМИЧЕСКИХ АППАРАТОВ**

**Александр Евгеньевич Привалов**

Военно-космическая академия имени А.Ф. Можайского,  
Санкт-Петербург, Россия  
vka@mil.ru

*Аннотация.* В статье предложена унифицированная программная платформа для разработки многоагентных моделей орбитальных группировок (ОГ) космических аппаратов (КА). Разработаны математическая и объектно-ориентированная модели программной платформы, особенностью которых является высокий уровень абстрагирования и расширенное множество систем координат, что позволяет рассматривать разработанные модели в качестве платформы для разработки многоагентных моделей ОГ КА различного целевого назначения с различной степенью детализации. Предложен вариант реализации многоагентной модели ОГ КА на языке Python с применением унифицированной программной платформы, обеспечивающий единство структуры, методов и алгоритмов соподчиненных и

взаимодействующих моделей ОГ КА различного целевого назначения на всех стадиях жизненного цикла.

**Ключевые слова:** многоагентная модель, орбитальная группировка, объектно-ориентированная модель, программная платформа

**Для цитирования:** Привалов А.Е. Унифицированная программная платформа для разработки многоагентных моделей орбитальных группировок космических аппаратов // Труды МАИ. 2022. № 123. DOI:[10.34759/trd-2022-123-22](https://doi.org/10.34759/trd-2022-123-22)

## UNIFIED SOFTWARE PLATFORM FOR DEVELOP OF MULTI-AGENT MODELS OF ORBITAL SPACECRAFTS CONSTELLATION

**Aleksandr E. Privalov**

Military space Academy named after A.F. Mozhaisky,

Saint Petersburg, Russia

[vka@mil.ru](mailto:vka@mil.ru)

**Abstract.** The article proposes a unified software platform designated for multi-agent models developing of orbital spacecraft constellation with various target purpose. Mathematical model of the software platform was developed at a high abstraction level, ensuring a generalized description of orbital spacecraft constellation operation, regardless its target purpose and life cycle stage. Besides, the basic systems of coordinates, employed in the spacecraft ballistics, as well as function of transition between them were defined in the space of the model functioning. These specifics allow considering this mathematical

model as a basis for the multi-agent models of the orbital spacecraft constellations of various target purposes. The object-oriented model based on the mathematical model in the form classes diagram, ensuring the platform program realization, was developed in the UML language

The software platform has realized in the Python programming language in the form of two packages, containing a set of classes and methods for developing multi-agent models of orbital spacecraft constellation with certain target purpose. Besides the program platform, such model should include the following modules: user classes library, which are the platform subsidiary classes, reflecting the target purpose of the spacecraft constellation; the experiment control subsystem, and information analysis and display subsystem.

The unified software platform may be implemented as a core for the multi-agent models developing of orbital spacecraft constellations with various target purposes. A high abstraction level allows creating models with various degrees of detail, corresponding to the life cycle stage of the orbital spacecraft constellation, keeping herewith the unity of the model, methods and algorithms.

**Keywords:** multi-agent model, orbital spacecraft constellation, object-oriented model, software platform

**For citation:** Privalov A.E. Unified software platform for develop of multi-agent models of orbital spacecrafts constellation. *Trudy MAI*, 2022, no. 123. DOI:[10.34759/trd-2022-123-22](https://doi.org/10.34759/trd-2022-123-22)

## Введение

Одной из характерных особенностей современного этапа освоения космического пространства является существенный рост количества космических аппаратов (КА) в составе орбитальных группировок (ОГ). В настоящее время разворачиваются такие ОГ, как Starlink и OneWeb [1-3], разрабатывается ОГ Сфера [4-5]. Для разработки и эксплуатации современных ОГ КА повсеместно внедряется CALS-технология, необходимым элементом которой является комплекс цифровых (математических и компьютерных) моделей [6-8]. Уровень развития последних в настоящее время позволяет добиться высокой степени адекватности описания структуры, функциональности и поведения ОГ на всех стадиях их жизненного цикла. Объединение этих моделей позволяют получить новую сущность – цифровой двойник, под которым понимается система, состоящая из цифровой модели изделия и двусторонних информационных связей с изделием и (или) его составными частями [9]. Основным элементом цифрового двойника является имитационная модель объекта [10-13]. Многоспутниковая ОГ КА как объект моделирования представляет собой распределенную многоагентную робототехническую систему, структуру, функциональность и поведение которой наиболее полно описывают многоагентные модели. В настоящее время в области разработки многоагентных моделей ОГ КА достигнуты значительные успехи [14-18]. Несмотря на то, что разработанные модели ориентированы на ОГ различного назначения, они имеют достаточно много общих черт. Во первых, структура перечисленных моделей определена типовой структурой многоагентной модели: модели агентов, среды их функционирования, а также модели

взаимодействия агентов между собой и со средой. Во вторых, все модели используют одни и те же методы и алгоритмы баллистики КА. В связи с этим, актуальной является разработка унифицированной программной платформы для разработки многоагентных моделей (УПП) ОГ КА, предназначенной для разработки на её основе многоагентных моделей ОГ различного целевого назначения. Отличительной особенностью УПП является высокий уровень абстракции и реализацию общих методов и алгоритмов баллистики КА, которые обеспечат единство соподчиненных и взаимодействующих моделей ОГ КА различного целевого назначения на всех стадиях жизненного цикла.

### **Постановка задачи**

Исходными данными для разработки УПП ОГ КА являются:

- множество КА и параметры их орбит;
- множество наземных пунктов управления (НПУ) и координаты их расположения;
- момент времени начала моделирования.

Требуется разработать многоагентную модель ОГ, обеспечивающий выполнение следующих функций:

- определение параметров движения агентов (КА и НПУ) в любой момент времени в любой из используемых в баллистике КА системе координат;
- определение условий взаимодействия между агентами в любой момент времени;

– расширение функционала агентов без внесения изменений в разработанный программный продукт.

### **Математическая модель унифицированной программной платформы**

Математическая модель УПП представлена кортежем

$$\Upsilon = \langle T, S, \Theta, M \rangle.$$

Рассмотрим элементы кортежа.  $T$  – непрерывное множество моментов времени ( $T \in [0, \infty)$ ).

$S$  – множество систем координат (СК), принадлежащих к одному из четырех классов [19]

$$S = \{S_1, S_2, S_3, S_4\}.$$

Рассмотрим элементы множества  $S$ .  $S_1$  – класс декартовых СК, к которому относятся абсолютная геоцентрическая экваториальная СК (АГЭСК), гринвичская СК, орбитальная СК и др.

$$S_1 = \left\{ S_{1j} = \left\{ (x_j, y_j, z_j, dx_j, dy_j, dz_j) \right\} \middle| j = \overline{1, m_1} \right\},$$

где  $S_{1j}$  –  $j$ -я декартовая СК,  $m_1$  – количество декартовых СК в определенных в УПП,  $x_j, y_j, z_j$  – декартовы координаты точки в  $j$  – й СК,  $dx_j, dy_j, dz_j$  – проекции скорости движения точки на оси координат  $j$  – й СК.

$S_2$  – класс сферических СК, к которому относятся географическая СК, стартовая сферическая и др.

$$S_2 = \left\{ S_{2j} = \left\{ \left( \varphi_j, \psi_j, r_j, d\varphi_j, d\psi_j, dr_j \right) \right\} \middle| j = \overline{1, m_2} \right\},$$

где  $S_{2j}$  –  $j$ -я сферическая СК,  $m_2$  – количество сферических СК определенных в УПП,  $\varphi_j$  – зенитный угол  $j$ -й СК,  $\psi_j$  – азимутальный угол  $j$ -й СК,  $r_j$  – радиус-вектор  $j$ -й СК,  $d\varphi_j, d\psi_j, dr_j$  – проекции скорости движения точки на оси координат  $j$ -й СК.

$S_3$  – класс цилиндрических СК, к которому относятся цилиндрическая геоцентрическая СК и др.

$$S_3 = \left\{ S_{3j} = \left\{ \left( \omega_j, r_j, z_j, d\omega_j, dr_j, dz_j \right) \right\} \middle| j = \overline{1, m_3} \right\},$$

где  $S_{3j}$  –  $j$ -я цилиндрическая СК,  $m_3$  – количество цилиндрических СК определенных в УПП,  $\omega_j$  – угол между полярной осью экваториальной плоскости и радиус-вектором  $j$ -й СК,  $r_j$  – радиус-вектор точки в экваториальной плоскости  $j$ -й СК,  $z_j$  – координата ортогональной проекции точки на зенитную ось  $j$ -й СК,  $d\omega_j, dr_j, dz_j$  – проекции скорости движения точки на оси координат  $j$ -й СК.

$S_4$  – класс оскулирующих элементов орбиты

$$S_4 = \left\{ S_{4j} \middle| j = \overline{1, m_4} \right\},$$

где  $S_{4j}$  –  $j$ -я система оскулирующих элементов. В связи с тем, что оскулирующие элементы могут иметь различный физический смысл, общей структуры для класса  $S_4$  не предлагается. В УПП используются кеплеровские элементы

$$S_{41} = \{(\Omega, i, \omega, a, e, \tau)\},$$

где  $\Omega$  – прямое восхождение восходящего узла орбиты,  $i$  – наклонение орбиты,  $\omega$  – аргумент широты перигея орбиты,  $a$  – большая полуось орбиты,  $e$  – эксцентриситет орбиты,  $\tau$  – время прохождения перигея.

$\Theta$  – матрица функций перехода от одной системы координат к другой

$$\Theta = \left( \theta_{jk} : S_j \times T \rightarrow S_k \mid S_j, S_k \in \bigcup_{l=1}^4 S_l \quad q = \sum_{l=1}^4 \text{card } S_l \right).$$

Элементами любой многоагентной модели являются модели агентов, модель среды, модели взаимодействия агентов между собой и со средой. Агентами в УПП являются КА, наземные пункты, а также динамические объекты, оказывающие влияние на остальные агенты. К качеству последних в УПП рассматриваются Солнце, Земля, Луна.

Модель агента-КА представляет собой кортеж

$$M_{KA} = \langle \kappa, Q, D, \eta, V, \pi \mid \pi = \langle O, \chi \rangle \rangle.$$

Здесь  $\kappa$  – кеплеровские элементы орбиты КА ( $\kappa \in S_4$ );

$Q$  – множество управляемых параметров КА ( $\kappa$  может содержаться в  $Q$ );

$D$  – множество доменов (здесь домен – область значений) управляемых параметров;

$\eta$  – отношение, ставящее в соответствие каждому параметру соответствующий домен

$$\eta = \left\{ \langle q_u, d_y \rangle \mid f_\eta(q_u, d_y) = 1 \right\}, \eta \subseteq Q \times D;$$

$$f_\eta(q_u, d_y) = \begin{cases} 1 & \text{— если } d_y \text{ является доменом для } q_u; \\ 0 & \text{— в противном случае;} \end{cases}$$

$V$  — множество векторов значений  $v_u \in R_n(q_u, \eta)$  управляемых параметров  $q_u \in Q$  (здесь  $R_n$  — левое сечение отношения  $\eta$  по параметру  $q_u$ , суть — домен параметра  $q_u$ )

$$V = \left\{ v = \left( \langle q_u, v_u \rangle \mid q_u \in Q, v_u \in R_n(q_u, \eta), u = \overline{1, \|Q\|} \right) \right\};$$

$\pi$  — программа функционирования КА;

$O$  — множество операций, выполняемых оборудованием КА

$$O = \left\{ o_j = \left\langle \tau_j^{(0)}, \tau_j, \mathcal{G}_j \right\rangle, j = \overline{1, m_O} \right\},$$

где  $\tau_j^{(0)}$  — начало  $j$ -й операции ( $\tau_j^{(0)} \in T$ ),

$\tau_j$  — окончание  $j$ -й операции ( $\tau_j \in T$ ),

$\mathcal{G}_j$  — функция, преобразующая начальный вектор  $V_j^{(0)}$  в вектор значений  $v_u \in S_n(q_u, \eta)$  управляемых параметров КА  $q_u \in Q$  в результате выполнения операции (функция, отражающая предназначение операции)

$$\mathcal{G}_j : V \times T \rightarrow V;$$

$\chi$  — функция, отражающая топологию программы функционирования  $\pi$

$$\chi = \left\{ \langle o_i, o_j \rangle \mid f_\chi(o_i, o_j) = 1 \right\}, \chi \subseteq O \times O;$$

$$f_\chi(o_i, o_j) = \begin{cases} 1 & \text{— если } o_i \text{ является предшественником } o_j; \\ 0 & \text{— в противном случае;} \end{cases}$$

Модель среды функционирования агентов как пространственно-временная область

$$E = S_A \times T,$$

где  $S_A \in S_1$  – АГЭСК.

Модель наземного пункта управления

$$M_{НПУ} = \langle g, R, W \rangle,$$

где  $g$  – географические координаты НПУ  $g \in S_g$ , где  $S_g \in S_2$  – система географических координат,

$R$  – множество параметров НПУ,

$W$  – множество значений параметров  $R$

$$W = \left\{ \langle r_u, w_u \rangle \mid u = \overline{1, \|R\|} \right\}.$$

В связи с тем, что параметры НПУ полагаются неизменными в процессе функционирования модели, включение доменов параметров в состав модели полагается излишним.

Модель движения Земли

$$M_3 = \langle T, A, f_\alpha \rangle,$$

где  $A$  – множество углов поворота Земли вокруг своей оси относительно положения в момент времени  $t = 0$   $A = [0, 2\pi)$ ,

$f_\alpha$  – функция моделирования вращения Земли

$$\alpha = f_\alpha(t), \alpha \in A, t \in T.$$

Модель движения Луны

$$M_L = \langle T, S_A, f_\mu \rangle,$$

где  $f_\mu$  – функция моделирования движения Луны

$$s_\mu = f_\mu(t), s_\mu \in S_A, t \in T.$$

Модель движения Солнца

$$M_C = \langle T, S_A, f_\xi \rangle,$$

где  $f_\xi$  – функция моделирования движения Солнца

$$s_\xi = f_\xi(t), s_\xi \in S_a, t \in T.$$

Многоагентная модель ОГ КА имеет вид

$$M = \langle E, \{M_{KA}\}, \{M_{HKV}\}, M_3, M_L, M_C, \Lambda \rangle.$$

где  $\Lambda$  – матрица функций взаимодействия между объектами

$$\Lambda = \left( \lambda_{jk}(m_i, m_j) \middle| m_i, m_j \in \{ \{M_{KA}\}, \{M_{HKV}\}, M_3, M_L, M_C \} \right).$$

Каждая функция взаимодействия зависит от типа объектов. Например, функция взаимодействия между Солнцем и КА состоит в условии освещения КА. Функции взаимодействия между активными объектами: КА и НПУ более сложные и зависят от целевого назначения КА. В рамках УПП целесообразно определить только шаблоны функций взаимодействия, которые будут конкретизироваться в процессе создания имитационной модели.

Математическая модель разработана на высоком уровне абстрагирования обеспечивающем обобщенное описание процесса функционирования ОГ КА, независимо от ее целевого назначения и стадии жизненного цикла. Кроме того, в пространстве функционирования модели определены основные системы координат,

используемые в баллистике космических аппаратов, а также функции перехода между ними. Данные особенности позволяют рассматривать разработанную модель в качестве платформы для разработки многоагентных моделей ОГ КА различного целевого назначения.

### **Объектно-ориентированная модель унифицированной программной платформы**

Объектно-ориентированная модель УПП проектировалась в соответствии с принципами языка объектно-ориентированного моделирования UML [21]. В соответствии с разработанной математической моделью были выделены классы, которые были условно разделены на две группы:

1. Классы, описывающие системы координат  $S$  и модельное время  $T$  (рис. 1).
2. Классы, описывающие многоагентную модель  $M$  (рис. 2).

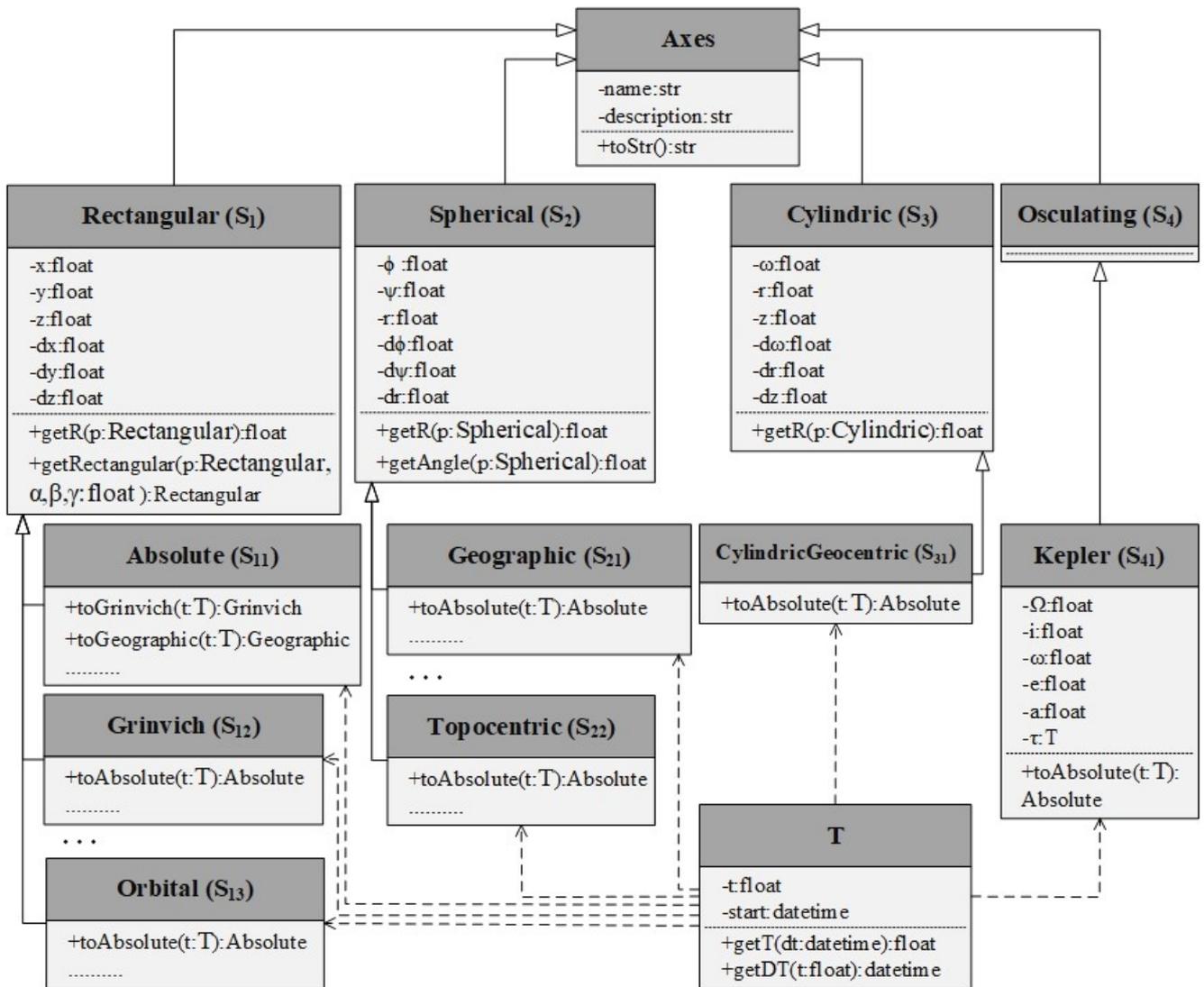


Рис. 1. Классы, описывающие системы координат и модельное время

Каждый момент времени моделирования является экземпляром класса  $T$  который содержит два атрибута:  $t \in [0, \infty)$  – модельное время, измеряемое в секундах от начала моделирования;  $start$  – начальный момент времени в формате *datetime*, который определяет дату и время начала моделирования. Кроме того, в классе  $T$  определены две функции преобразования модельного времени в реальное (*getDT*) и обратно (*getT*).

Для описания системы координат введен базовый класс *Axes*, содержащий атрибуты описательного характера *name* и *description*, а также метод *toStr()*, возвращающий строку со значениями всех атрибутов класса. Дочерними для класса *Axes* являются классы *Rectangular*, *Spherical*, *Cylindric*, *Osculating*, определяющие прямоугольные  $S_1$ , сферические  $S_2$ , цилиндрические  $S_3$ , оскулирующие  $S_4$  СК соответственно. Для каждого класса (кроме *Kepler*) определена функция *getR(p)* с параметром *p*, который задает координаты точки в той же СК. Функция возвращает расстояние между двумя точками (текущей и переданной через *p*) в данной СК. Для прямоугольной СК определен метод *getRectangular(p:Rectangular,α,β,γ:float)*, который возвращает координаты точки в прямоугольной СК с началом координат, заданным параметром *p*, оси которой повернуты вокруг осей *x*, *y*, *z* на углы  $\alpha$ ,  $\beta$ ,  $\gamma$  соответственно. Для сферической СК определен метод *getAngle(p)*, возвращающий угловое расстояние между двумя точками. Следует отметить, что множество типовых задач, решаемых для различных СК, достаточно обширно. В связи с этим, библиотека методов для каждого класса должна пополняться по мере необходимости в процессе развития УПП.

Для каждого из классов второго уровня определены дочерние классы, определяющие СК, используемые в баллистических расчетах. На диаграмме приведены лишь некоторые из используемых систем координат: *Absolute* – АГЭСК, *Grinvich* – гринвичская, *Orbital* – орбитальная, *Geographic* – географическая, *Topocentric* – топоцентрическая сферическая, *CylindricGeocentric* – цилиндрическая геоцентрическая, *Kepler* – координаты в кеплеровских элементах. Матрица  $\Theta$

функций перехода от одной системы координат к другой реализована методами классов второго уровня. Например, для класса *Absolute* (АГЭСК) это могут быть методы: *toGrinvich(t:T)*, возвращающий координаты точки в гринвичской СК для заданного момента времени *t*, *toGeographic(t:T)* – в географической сферической СК и т.д.

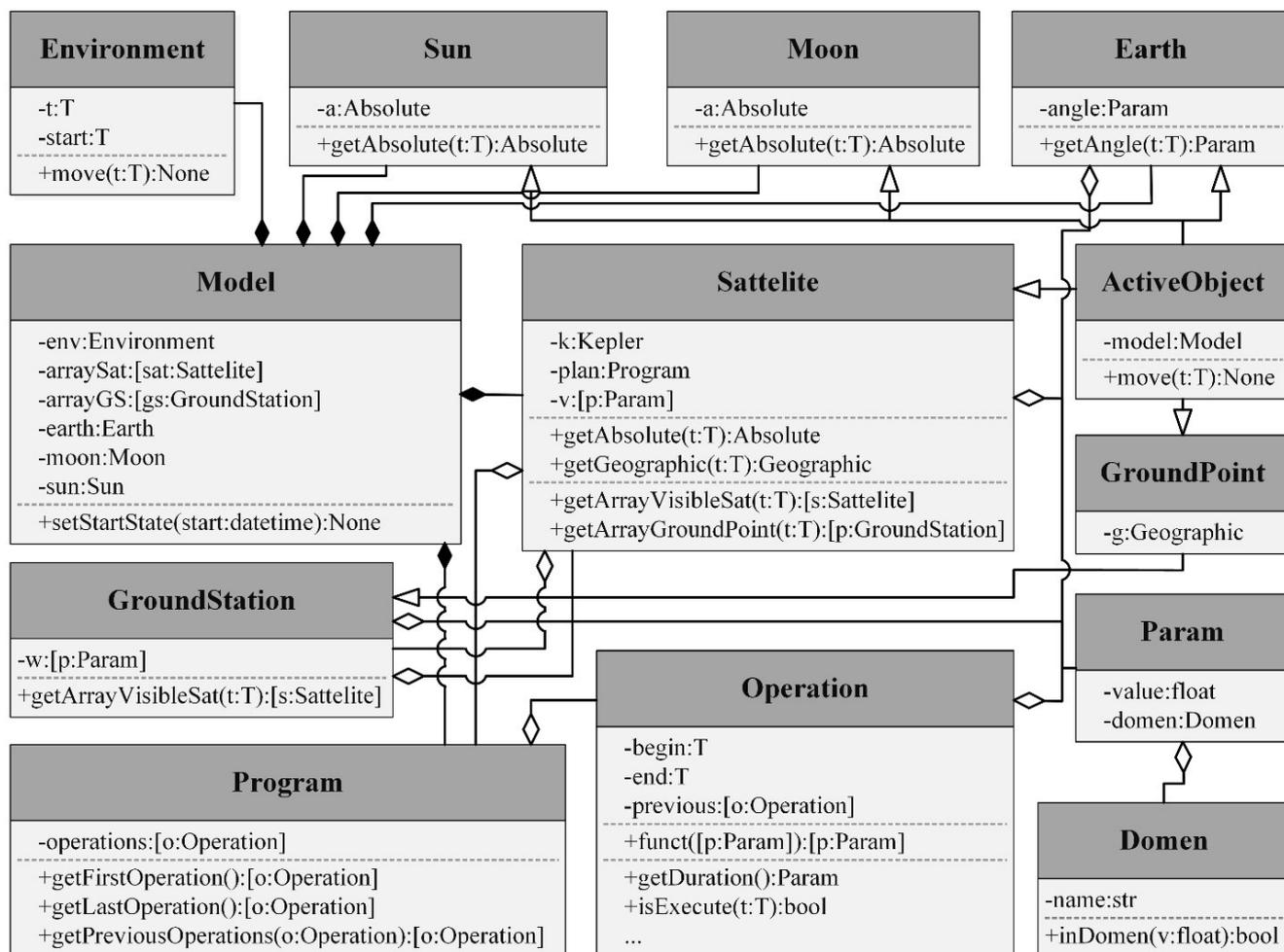


Рис. 2. Классы, описывающие многоагентную модель

Корневым классом многоагентной модели является класс *Model*, атрибутами которого является множество экземпляров классов активных объектов. Для активных объектов определен класс *ActiveObject*, который является родительским для элементов многоагентной модели. Для него определен атрибут *model* класса *Model*,

определяющий корневой объект многоагентной модели, а также метод *move*, изменяющий состояние активного объекта во внешней среде. Таким образом, класс *ActiveObject* связан с классом *Model* отношением агрегирования, вместе с тем, имеет место обратное отношение зависимости. Дочерними классами для *ActiveObject* являются:

1. Класс *Sattelite* ( $M_{KA}$ ) – модель КА. Для него определены атрибуты:  $\kappa \in S_4$  – кеплеровские элементы орбиты, *plan* ( $\pi$ ) – программа функционирования бортового оборудования, и  $v \in V$  – вектор текущих параметров КА. Для класса *Sattelite* определены следующие функции: *getAbsolute*( $t:T$ ) – функция расчета и прогнозирования параметров движения центра масс в АГЭСК, *getGeographic*( $t:T$ ) – функция расчета и прогнозирования параметров движения центра масс в географической СК, *getArrayVisibleSat*( $t:T$ ) – функция, возвращающая список КА, находящихся в зоне радиовидимости в требуемый момент времени, *getArrayGroundPoint*( $t:T$ ) – функция, возвращающая список наземных станций, в зоне радиовидимости которых находится КА. Последние две функции являются элементами матрицы взаимодействия  $\Lambda$ .

2. Класс *GroundStation* ( $M_{HKV}$ ) – модель наземного пункта НПУ. В целях расширения функционала УПП возможностью работать с отдельными точками на поверхности Земли, определен класс точки *GroundPoint*, атрибутом которого являются географические координаты точки *g*. Данный класс является родительским для класса *GroundStation* и дочерним для класса *ActiveObject*. Для класса *GroundStation* определен вектор  $w \in W$  – вектор текущих параметров НПУ, а также

функция  $getArrayVisibleSat(t:T)$ , возвращающая список КА, находящихся в зоне радиовидимости НПУ.

3. Классы небесных тел  $Earth (M_3)$ ,  $Moon (M_{II})$ , и  $Sun (M_C)$ . Для каждого класса определен атрибут его состояния: для  $Earth$  – угол поворота  $angle (\alpha)$ , для  $Moon$  и  $Sun$  – координаты небесного тела в АГЭСК ( $s_\mu$  и  $s_\xi$ ). Кроме того, для каждого класса определен метод, возвращающий значение атрибутов состояния для заданного момента времени.

Для описания программы функционирования введены классы  $Program (\pi)$  и  $Operation (O)$ . Класс  $Program$  содержит атрибут  $operations$ , являющийся массивом элементов класса  $Operation$ , а также ряд сервисных функций, упрощающих работу с классом. К таким функциям относятся:  $getFirstOperation()$  и  $getLastOperation()$ , возвращающие массивы первых и последних операций программы соответственно,  $getPreviousOperations(o:Operation)$ , возвращающая массив предшествующих операция для заданной и другие функции.

Атрибутами класса  $Operation$  являются:  $begin (\tau_j^{(0)})$  – начало операции,  $end (\tau_j)$  – окончание операции, а также массив предшествующих операций  $pervious$ , реализующий отношение  $\chi$ . Для класса определен метод  $funct([p:Param])$ , реализующий отношение  $\mathcal{A}_j$ , который возвращает массив значений параметров после выполнения операции. Кроме того, введены ряд сервисных методов, упрощающих работу с экземплярами классов, таких, как  $getDuration()$ , возвращающий

длительность операции, или  $isExecute(t:T)$ , возвращающий признак выполнения операции в заданное время.

Все параметры модели являются экземплярами класса *Param*, который содержит в качестве атрибута область определения параметра *domen*, принадлежащего классу *Domen*. Модель среды функционирования агентов определяется классом *Environment*, функцией которой является продвижение модельного времени. Для этого определены атрибуты  $t$  – текущее время, и  $start$  – начальное время, а также метод  $move(t:T)$ , изменяющий текущее время.

Таким образом, на основе математической модели разработана объектно-ориентированная модель в виде диаграммы классов на языке UML, обеспечивающая программную реализацию УПП.

### **Разработка многоагентных моделей орбитальных группировок космических аппаратов**

Унифицированная программная платформа для разработки многоагентных моделей ОГ КА реализована в виде двух пакетов на языке Python [21] *coordinate\_pack* и *agent\_pack*, структура которых представлена на рис. 3. Пакет *coordinate\_pack* может использоваться автономно для решения различных задач баллистики КА, для использования *agent\_pack* необходим пакет *coordinate\_pack*. Кроме того, для функционирования УПП необходимы следующие библиотеки языка Python: *numpy* [22], *scipy* [23], *datetime* [24].

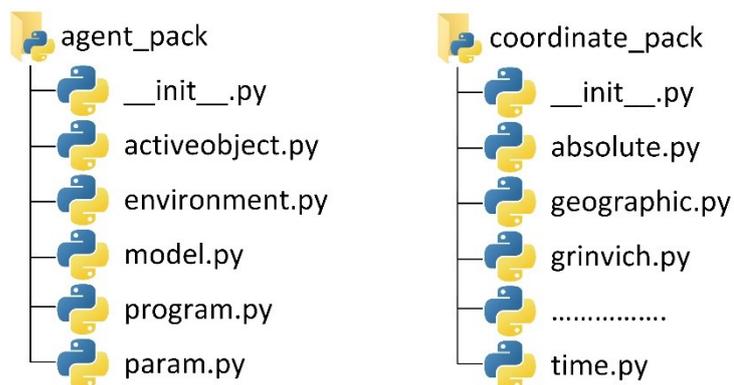


Рис. 3. Структура модулей УПП

Унифицированная программная платформа не предназначена для автономного функционирования, а содержит совокупность классов и методов для разработки многоагентных моделей ОГ определенного целевого назначения. Предлагается следующая структура такой модели, представленная на рис 4. В соответствии с целевым назначением моделируемой ОГ КА разрабатывается библиотека пользовательских классов, являющихся дочерними классам УПП. Кроме того, в соответствии с целью разработки модели на основе классов УПП разрабатывается подсистема управления экспериментом. Многоагентная модель ОГ КА строится из экземпляров классов вышеперечисленных трех компонентов. На подсистему отображения информации выводятся как непосредственно результаты функционирования многоагентной модели, так и результаты их анализа (в том числе статистического), выполняемого соответствующей подсистемой.

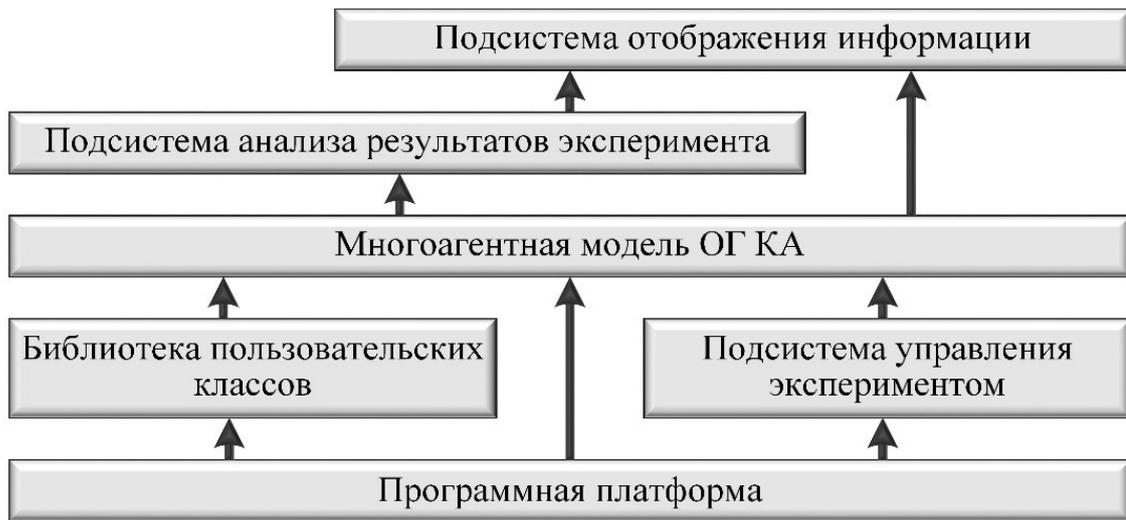


Рис. 4. Структура многоагентной модели ОГ КА

### Заключение

Традиционно, применение многоагентных моделей ограничивалось стенами научных лабораторий и институтов, где решались задачи исследования и обоснования характеристик ОГ КА. Вместе с тем, особенности современного этапа развития космонавтики, такие, как многоспутниковые ОГ, серийное производство КА, CALS-технологии и применение цифровых двойников ставят перед многоагентными моделями ОГ КА новые задачи, связанные с сопровождением ОГ КА на каждом этапе жизненного цикла, управлением ОГ в реальном масштабе времени, прогнозированием ее состояния и функционирования и т.д. Многообразие применяемых моделей на различных стадиях жизненного цикла для различных ОГ КА снижает эффективность их применения и вызывает необходимость обеспечения совместимости соподчиненных и взаимодействующих моделей.

Разработанная в рамках исследований и представленная в настоящей статье унифицированная программная платформа может быть использована в качестве ядра

для разработки многоагентных моделей ОГ КА различного целевого назначения. Высокий уровень абстрагирования позволяет создавать на ее основе модели различной степени детализации, соответствующей стадии жизненного цикла ОГ КА, сохраняя при этом единство структуры модели, методов и алгоритмов.

### **Список источников**

1. Мырова Л.О., Ментус О.В., Давыдов А.Б. и др. Низкоорбитальные спутниковые системы связи Starlink и OneWeb // Труды Научно-исследовательского института радио. 2021. №2. С. 36-45. DOI: [10.34832/NIIR.2021.5.2.005](https://doi.org/10.34832/NIIR.2021.5.2.005)
2. Carlos Álvaro Arroyo Parejo, Noelia Sánchez Ortiz, Raúl Domínguez González. Effect of mega-constellations on collision risk in space // Proc. 8<sup>th</sup> European Conference on Space Debris (virtual), Darmstadt, Germany, 20-23 April 2021, URL: <http://conference.sdo.esoc.esa.int>
3. Harrison Krantz, Eric C. Pearce, Adam Block. Characterizing the All-Sky Brightness of Satellite Mega-Constellations and the Impact on Astronomy Research // Proceedings of the 2021 AMOS Conference. URL: <https://arxiv.org/ftp/arxiv/papers/2110/2110.10578.pdf>
4. Афанасьев И. «Сфера» общих интересов // Русский космос. 2020. № 8. С. 8-19. URL: <https://www.roscosmos.ru/media/pdf/russianspace/rk2020-08-single.pdf>
5. Стельмащук А.С., Набойченко С.А. Низкоорбитальные спутниковые системы: настоящее состояние развития на примере сравнения Starlink и Сфера // III Всероссийская научно-техническая конференция «Состояние и перспективы развития современной науки по направлению «АСУ, информационно-

телекоммуникационные системы» (Анапа, 22–23 апреля 2021): сборник статей. – Анапа: Военный инновационный технополис "ЭРА", 2021. С. 68-71.

6. Заковряшин А.И. ИПИ технология создания наукоемких изделий // Труды МАИ. №

49. URL: [http://trudymai.ru/published.php?ID=28072&PAGEN\\_2=2](http://trudymai.ru/published.php?ID=28072&PAGEN_2=2)

7. Бадалов А.Ю., Разумов Д.А. Методика моделирования в жизненном цикле большой автоматизированной системы космодрома уровня Smart City // Труды МАИ. 2018. №

100. URL: <http://trudymai.ru/published.php?ID=93491>

8. Сазонов А.А., Джамай В.В., Повеквечных С.А. Анализ эффективности внедрения CALS технологий (на примере отечественного авиастроения) // Организатор производства. 2018. Т. 26. № 1. С. 84-92. DOI: [10.25065/1810-4894-2018-26-1-84-92](https://doi.org/10.25065/1810-4894-2018-26-1-84-92)

9. ГОСТ Р 57700.37-2021. Компьютерные модели и моделирование. Цифровые двойники изделий. Общие положения. - М.: Российский институт стандартизации, 2021. 16 с.

10. Гусев П.Ю. Автоматизация планирования производственных процессов авиастроительного предприятия с использованием цифрового двойника // Труды МАИ. 2018. № 103. URL: <http://trudymai.ru/published.php?ID=101190>

11. Петров А.В. Имитационное моделирование как основа технологии цифровых двойников // Вестник Иркутского государственного технического университета. 2018. Т. 22. № 10. С. 56–66. DOI: [10.21285/1814-3520-2018-10-56-66](https://doi.org/10.21285/1814-3520-2018-10-56-66)

12. Балухто А.Н. Соколов Б.В., Карсаев О.В. Облачная платформа iwebsim как средство имитационного моделирования космических систем // Десятая всероссийская научно-практическая конференция по имитационному

моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» (ИММОД-2021): труды конференции. - СПб.: Центр технологии судостроения и судоремонта, 2021. С. 95-104.

13. Glaessgen E.H., Stargel D.S. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles // Paper for the 53rd Structures, Structural Dynamics, and Materials Conference: Special Session on the Digital Twin. 2012. URL: <https://ntrs.nasa.gov/api/citations/20120008178/downloads/20120008178.pdf>.

DOI:[10.2514/6.2012-1818](https://doi.org/10.2514/6.2012-1818)

Соллогуб А.В., Скобелев П.О., Симонова Е.В. и др. Мультиагентные технологии распределённого управления группировкой малоразмерных космических аппаратов дистанционного зондирования Земли // Информационное общество. 2013. № 1–2. С. 58 – 69.

15. Городецкий В.И., Карсаев О.В. Самоорганизация группового поведения кластера малых спутников распределенной системы наблюдения // Известия ЮФУ. Технические науки. 2017. № 187. С. 234–247. DOI: [10.18522/2311-3103-2017-1-234247](https://doi.org/10.18522/2311-3103-2017-1-234247)

16. Карсаев О.В. Имитационное моделирование автономного управления группировкой малых спутников // Известия ЮФУ. Технические науки. 2018. № 1 (195). С. 140–154. DOI: [10.23683/2311-3103-2018-1-140-154](https://doi.org/10.23683/2311-3103-2018-1-140-154)

17. Фоминов И.В., Привалов А.Е. Ключевые проблемы развития группового управления малыми космическими аппаратами дистанционного зондирования Земли // Космонавтика и ракетостроение. 2019. № 1 (106). С. 31-35.

18. Привалов А.Е., Федяев В.В., Бугайченко П.Ю. Применение многоагентных технологий для построения имитационных моделей многоспутниковых орбитальных группировок дистанционного зондирования Земли // Труды конференции «Имитационное моделирование систем военного назначения ИМСВН – 2020». - СПб: ВА МТО – АО «ЦТСС», 2021. С. 219–228.
19. Нариманов Г.С., Тихонравов М.К. Основы теории полета космических аппаратов. - М.: Машиностроение, 1972. – 608 с.
20. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. - М.: ДМК Пресс, 2006. - 496 с.
21. Федоров Д.Ю. Программирование на языке высокого уровня Python. - М.: Издательство Юрайт, 2021. – 2010 с.
22. Numpy User Guide. Release 1.22.0. URL: <https://numpy.org/doc/stable/numpy-user.pdf>
23. Numpy and Scipy Documentation. URL: <https://docs.scipy.org/doc>
24. Datetime – Basic date and time types. Python Software Foundation, 2001-2022. URL: <http://docs.python.org/3/library/datetime.html#module-datetime>

## References

1. Myrova L.O., Mentus O.V., Davydov A.B. et al. *Trudy Nauchno-issledovatel'skogo instituta radio*, 2021, no 2, pp. 36-45. DOI: [10.34832/NIIR.2021.5.2.005](https://doi.org/10.34832/NIIR.2021.5.2.005)
2. Carlos Álvaro Arroyo Parejo, Noelia Sánchez Ortiz, Raúl Domínguez González. Effect of mega-constellations on collision risk in space, *Proc. 8th European Conference on Space*

- Debris (virtual)*, Darmstadt, Germany, 20-23 April 2021, URL: <http://conference.sdo.esoc.esa.int>
3. Harrison Krantz, Eric C. Pearce, Adam Block. Characterizing the All-Sky Brightness of Satellite Mega-Constellations and the Impact on Astronomy Research, *Proceedings of the 2021 AMOS Conference*. URL: <https://arxiv.org/ftp/arxiv/papers/2110/2110.10578.pdf>
4. Afanas'ev I. *Russkii kosmos*, 2020, no. 8, pp. 8-19. URL: <https://www.roscosmos.ru/media/pdf/russianspace/rk2020-08-single.pdf>
5. Stel'mashchuk A.S., Naboichenko S.A. *III Vserossiiskaya nauchno-tekhnicheskaya konferentsiya «Sostoyanie i perspektivy razvitiya sovremennoi nauki po napravleniyu «ASU, informatsionno-telekommunikatsionnye sistemy»: sbornik statei*. Anapa, Voennyi innovatsionnyi tekhnopolis "ERA", 2021, pp. 68-71.
6. Zakovryashin A.I. *Trudy MAI*, no. 49. URL: [http://trudymai.ru/eng/published.php?ID=28072&PAGEN\\_2=2](http://trudymai.ru/eng/published.php?ID=28072&PAGEN_2=2)
7. Badalov A.Yu., Razumov D.A. *Trudy MAI*, 2018, no. 100. URL: <http://trudymai.ru/eng/published.php?ID=93491>
8. Sazonov A.A., Dzhamai V.V., Povekvechnykh S.A. *Organizator proizvodstva*, 2018, vol. 26, no. 1, pp. 84-92. DOI: [10.25065/1810-4894-2018-26-1-84-92](https://doi.org/10.25065/1810-4894-2018-26-1-84-92)
9. *GOST R 57700.37-2021. Komp'yuternye modeli i modelirovanie. Tsifrovye dvoyniki izdelii. Obshchie polozheniya* (State Standard 57700.37-2021. Computer models and simulation. Digital twins of products. General provisions), Moscow, Rossiiskii institut standartizatsii, 2021, 16 p.

10. Gusev P.Yu. *Trudy MAI*, 2018, no. 103. URL: <http://trudymai.ru/eng/published.php?ID=101190>
11. Petrov A.V. *Vestnik Irkutskogo gosudarstvennogo tekhnicheskogo universiteta*, 2018, vol. 22, no 10, pp. 56–66. DOI: [10.21285/1814-3520-2018-10-56-66](https://doi.org/10.21285/1814-3520-2018-10-56-66)
12. Balukhto A.N. Sokolov B.V., Karsaev O.V. *Desyataya vserossiiskaya nauchno-prakticheskaya konferentsiya po imitatsionnomu modelirovaniyu i ego primeneniyu v nauke i promyshlennosti «Imitatsionnoe modelirovanie. Teoriya i praktika» (IMMOD-2021): trudy konferentsii*, Saint Petersburg, Tsentr tekhnologii sudostroeniya i sudoremonta, 2021, pp. 95-104.
13. Glaessgen E.H., Stargel D.S. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles, *Paper for the 53rd Structures, Structural Dynamics, and Materials Conference: Special Session on the Digital Twin*, 2012, URL: <https://ntrs.nasa.gov/api/citations/20120008178/downloads/20120008178.pdf>. DOI: [10.2514/6.2012-1818](https://doi.org/10.2514/6.2012-1818)
14. Sollogub A.V., Skobelev P.O., Simonova E.V. et al. *Informatsionnoe obshchestvo*, 2013, no. 1–2, pp. 58 – 69.
15. Gorodetskii V.I., Karsaev O.V. *Izvestiya YuFU. Tekhnicheskie nauki*, 2017, no. 187, pp. 234–247. DOI: [10.18522/2311-3103-2017-1-234247](https://doi.org/10.18522/2311-3103-2017-1-234247)
16. Karsaev O.V. *Izvestiya YuFU. Tekhnicheskie nauki*, 2018, no. 1 (195), pp. 140–154. DOI: [10.23683/2311-3103-2018-1-140-154](https://doi.org/10.23683/2311-3103-2018-1-140-154)
17. Fominov I.V., Privalov A.E. *Kosmonavtika i raketostroenie*, 2019, no. 1 (106), pp. 31-35.

18. Privalov A.E., Fedyaev V.V., Bugaichenko P.Yu. *Trudy konferentsii «Imitatsionnoe modelirovanie sistem voennogo naznacheniya IMSVN – 2020»*, Saint Petersburg, VA MTO – AO «TsTSS», 2021, pp. 219–228.
19. Narimanov G.S., Tikhonravov M.K. *Osnovy teorii poleta kosmicheskikh apparatov* (Basics of Spacecraft Flight Theory), Moscow, Mashinostroenie, 1972, 608 p.
20. Buch G., Rambo D., Yakobson I. *Yazyk UML. Rukovodstvo pol'zovatelya* (The Unified Modeling Language User Guide), Moscow, DMK Press, 2006, 496 p.
21. Fedorov D.Yu. *Programmirovaniye na yazyke vysokogo urovnya Python* (Python High-Level Programming: A textbook for universities), Moscow, Izdatel'stvo Yurait, 2021, 2010 p.
22. *NumPy User Guide. Release 1.22.0*. URL: <https://numpy.org/doc/stable/numpy-user.pdf>
23. *Numpy and Scipy Documentation* URL: <https://docs.scipy.org/doc>
24. *Datetime - Basic date and time types. Python Software Foundation, 2001-2022*. URL: <http://docs.python.org/3/library/datetime.html#module-datetime>

Статья поступила в редакцию 20.01.2022; одобрена после рецензирования 31.01.2022; принята к публикации 20/04.2022.

The article was submitted on 20.01.2022; approved after reviewing on 31.01.2022; accepted for publication on 20.04.2022.