УДК 629.78

ЯЗЫК РАЗРАБОТКИ ПЛАНА ПОЛЁТА, СОСТАВЛЯЕМОГО АВТОМАТИЗИРОВАННЫМ СПОСОБОМ ДЛЯ АВТОМАТИЧЕСКИХ КОСМИЧЕСКИХ АППАРАТОВ

Жигастова О.К.*, Почукаев В.Н.

Центральный научно-исследовательский институт машиностроения, ЦНИИмаш, ул. Пионерская, 4, Королёв, Московская область, 141070, Россия
** e-mail: inolga-ok@yandex.ru*

Рассмотрены принципы построения и структура языка описания командных конструкций, разработанного для подготовки исходных данных плана полёта. Представленные языковые конструкции позволяют формализовать структуру плана и составляющих его элементов, упростив процедуру планирования. Определёны синтаксис и семантика разработанного языка, выбраны лучшие варианты построения его конструкций, которые вошли в методику автоматизированного планирования. Опыт применения предложенного языка при создании автоматизированных систем планирования для автоматических космических аппаратов (КА) позволил повысить производительность и уменьшить количество ошибок при составлении плана, а также упростить ввод новых командных конструкций в план. Эффективность использования языка была доказана при управлении пятью автоматическими КА научного и социально-экономического назначения.

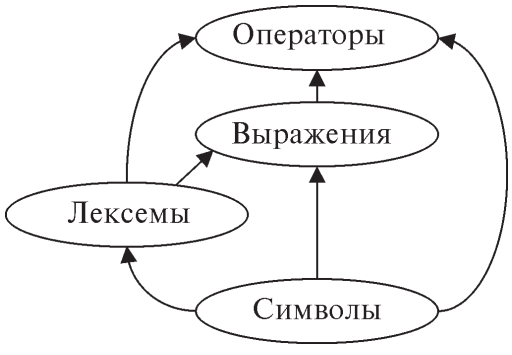
Ключевые слова: автоматический космический аппарат, планирование полёта, план полёта, командные конструкции, язык планирования полёта, методика планирования полёта, автоматизация планирования.

План полёта автоматического КА представляет собой программу, которая содержит множество данных, определяющих последовательность действий устройств и систем КА и наземного автоматизированного комплекса управления (НАКУ), в совокупности обеспечивающих реализацию составленного плана и целевой задачи управления КА.

Составление плана полета выполняется на программном комплексе автоматизированного планирования полёта [1] в Центре управления полетами (ЦУП). План полета формируется на основе исходных данных [2], подготовленных специалистами группы управления. Исходные данные берутся из плана работы целевой аппаратуры, плана пространственно-временных действий и совокупности данных о техническом состоянии автоматического КА и средств НАКУ, задействуемых при управлении КА. Исходные данные готовятся на специализированном языке, который в дальнейшем будем называть языком разработки плана полета. Этот язык использует синтаксис, основанный на синтаксисе языка программирования C++, содержит элементы языка C# и языковые конструкции, специфичные для задачи автоматизированного планирования полета автоматических КА.

В разрабатываемом для КА плане полёта, как и в программах для персонального компьютера, можно выделить несколько составляющих любого языка [3]: символы, лексемы (элементарные конструкции), выражения и операторы. Рассмотрим их более подробно применительно к решению задачи планирования полёта (см. рисунок).

Каждый элемент языка определяется синтаксисом и семантикой. Синтаксические определения устанавливают правила построения элементов языка, а семантика определяет их смысл и правила использования.



Структура алгоритмического языка

1. **Алфавит языка**, или его символы, — основные неделимые знаки, с помощью которых записываются конструкции языка [3]. Алфавит языка разработки плана полёта включает прописные и строчные буквы, арабские цифры, специальные знаки и пробельные символы.

2. **Лексема**, или элементарная конструкция [3], — минимальная единица языка. Она формируется из символов алфавита языка. Лексемы подразделяются на идентификаторы, ключевые (зарезервированные) слова, операции, константы, разделители.

Идентификатор — имя объекта плана полета. Например, к нему можно отнести тип одной из командных конструкций [4], используемых при составлении плана.

Ключевые слова — зарезервированные идентификаторы, которые имеют специальное назначение при составлении плана. Список основных ключевых слов приведён в табл. 1.

Таблица 1

Б	КИП	ПК	ТЕКСТ
В	КОС	ПП	ТП
ГИ	МК	Р	УВ
Д	НОС	РК	УП
ДВ	НОС/КОС	СР	Ф
ДПР	О	Т	Ц
З	ОВ	Т0	ЦК

Операции — один или два символа, определяющие действие над операндами. Операции представляют собой отдельные лексемы, используемые для действий над командами управления, стоящими в плане полета.

Константами называют неизменяемые величины. При составлении плана полета «константами» называются команды командных конструкций, которые обязательно должны быть реализованы в плане, они имеют идентификатор «*» — это «жёсткие» команды. Команды, не имеющие такого идентификатора, называются «мягкими».

Разделители — единичные символы, отделяющие лексемы друг от друга. К разделителям в плане полета относятся: скобки, запятая, точка с запятой, пробел, символ табуляции и символ конца строки.

Комментарии представляют собой обычный текст, стоящий после идентификатора «Текст». Комментарии используются для дачи пояснения к реализуемым в плане полета командным конструкциям или команде управления. При написании комментария можно использовать любые символы алфавита, кроме знаков операций и разделителей.

Для языка разработки плана полета было определено шесть типов данных: **int, char, bool, date, time, vector_p**. Первые три типа были наследованы из языка программирования C++ [3], последний — из C# [5].

В отличие от C++, для языка разработки плана полета типы данных не имеют отдельных зарезервированных идентификаторов, которые указывали бы на их тип. Тип данных задается автоматически по идентификатору, который определяет тип объекта плана полета.

Целый тип **int** используется для задания величины целого числа.

Символьный тип **char** предназначен для хранения любого символа из 256-символьного набора ASCII. Тип **char** используется для создания массивов, с помощью которых в структуру плана полета вписываются комментарии с текстом. Массив символов имеет постоянную длину 255 байт. Это означает, что одна строка комментария может содержать до 255 символов.

Логический тип **bool** может принимать значение только true и false. Внутренняя форма представления значения false — 0, любое другое значение интерпретируется как true.

Календарная дата, определенная как тип **date**, используется для хранения календарной даты в структуре «дд.мм.гггг».

Время **time** используется для хранения времени в структуре «чч:мм:сс».

Кортеж определен как тип **vector_p**. В языке программирования кортеж — это особый тип данных, как правило, безымянный, содержащий неизменный набор элементов, причем элементы в кортеже могут быть различных типов. К элементам кортежа можно только обращаться, а оперировать ими нельзя. Элементы кортежа не именованы. В языке разработки плана полета кортежи используются для хранения параметров команд управления, которые представляют собой вектор с параметрами [4], содержащий технологические данные, обеспечивающие реализацию команды средствами НАКУ и КА.

3. Для определения места команды в плане полёта, вычисления момента времени её выполнения в заданной последовательности команд используются **выражения** [3]. Выражение задаёт правило вычисления некоторого значения. Оно состоит из операндов (переменных) и знаков операций. Операнды задают данные для вычислений. Операции задают действия, которые необходимо выполнить.

Переменная. В каждой переменной хранятся данные определённого типа. У переменной есть имя и значение. Например, для каждой команды управления в плане полёта определён тип, свое название и значение времени, когда ее необходимо выполнить. Тип и имя команды служат для обращения к базе данных (БД) программного комплекса автоматизированного планирования полёта [1], где содержатся значения вектора ее параметров. Тип команды указывает, в какой таблице БД следует искать данную команду, а ее имя — в какой строке таблицы. Перед использованием любая команда должна быть описана в БД. Существует два типа команд управления [4]: программные команды (ПК), реализуемые на КА, и разовые команды (РК), реализуемые средствами НАКУ. Названия этих типов команд являются ключевыми словами и указаны в табл. 1. Для команд управления описание переменной будет следующее:

ПК | РК=*имя*(*значение*);

Ниже приведён общий вид оператора описания переменных:

Идентификатор=[*имя*](*значение1*, *значение2*,..., *значениеN*);

Для наглядности описания языковых конструкций здесь используется широко известный неформализованный способ, при котором текст, написанный по-русски, необходимо заменять конкретным значением, необязательные части заключаются в квадратные скобки, а выбор одного из нескольких элементов обозначается вертикальной чертой.

Для каждой переменной существует своя область действия. Областью действия переменной является блок, в котором она определена. Блоком в языке называется последовательность операторов, заключенная в фигурные скобки. Все команды следуют в блоке одна за другой.

При составлении плана полёта используются четыре вида блоков, в которых можно задавать переменные:

— *суточный интервал*, у которого область действия составляет одни календарные сутки;

— *витковый интервал*, область действия которого — один оборот КА вокруг Земли. Его длительность может составлять от 90 до 105 мин, в зависимости от орбиты КА;

— интервал *зоны радиовидимости* наземной станции, область действия — от 0 до «20 мин, которая также зависит от орбиты КА;

— *тело командной конструкции*, область действия — сама командная конструкция.

Операции. В табл. 2 приведены операции, определенные для действий, совершаемых с операндами. Операции используются для присвоения значения, для сложения и вычитания времени, выполнения условных операций.

Таблица 2

Операция	Назначение операции
=	присвоение
+	сложение
-	вычитание
+=	переход по исполнению
-=	переход по неисполнению

Операция присвоения используется в языке разработки плана полета как законченный оператор. Пример операции присвоения:

```
Б=(0);
```

Приведенное здесь выражение означает, что необходимо выполнить снятие блокировки команды управления на КА.

Арифметические операции сложения и вычитания используются для действий над временем исполнения команд управления. Рассмотрим командную конструкцию следующего вида:

```
Т0=(-00:15:48);
1)ПК=069*(+00:00:00);
2)ПК=071*(+00:00:02);
3)ПК=050*(+00:00:04);
```

Здесь приведено тело циклограммы «ЦГ18 "ЗГ1"» [4], обеспечивающей включение задающего генератора передатчика для выполнения операции по передаче накопленной научной информации с КА. Первая строка приведенного выражения означает, что данная циклограмма должна начать выполняться до того, как КА пересечет линию экватора на восходящей части витка. Идентификатор «Т0» здесь означает время пересечения экватора. Для того чтобы получить время начала работы циклограммы, необходимо из времени «Т0» вычесть 15 мин 48 с. Вторая строка означает, что команда ПК069 должна быть выдана на момент времени начала работы циклограммы. Третья — что команда ПК071 должна исполниться через 2 с после начала работы циклограммы. Для этого к времени начала прибавляется время исполнения команды. Следующая команда ПК050 должна выдаться через 4 с после выдачи команды ПК071. Для этого к уже вычисленному времени ПК071 прибавляется время команды ПК050.

Данная схема вычисления времени выдачи команд управления реализована не только для циклограмм, но и для других командных конструкций плана полета, кроме макрокоманд. В качестве базового значения времени «Т0» может использоваться не только время пересечения экватора, но и любые другие времена, например, время пересечения КА зоны радиосвязи с наземным командно-измерительным пунктом (КИП). Все базовые времена, используемые для выполнения расчетов, содержатся в плане пространственно-временных действий КА.

Операции перехода являются условными операциями и выполняются для задания в плане полета условий реализации команд управления на КИП. Пример операции перехода:

```
4)РК=165(+00:00:02);ДВ=(1);
5)РК=118(+00:00:02);ДВ=(1);
6)РК=122(+00:00:02);ДВ=(1);
7)РК=165(+00:00:02);ДВ=(1);
8)РК=125(+00:00:02);ДВ=(1);УВ+=3(5);З=(6);
9)РК=122(+00:00:02);ДВ=(1);
```

В представленном фрагменте рассматривается операция получения с КА в сеансе связи массива информации оперативного контроля «ИОК2» по команде РК125, выполняемой при условии успешной передачи на КА плана полета. Эта операция реализуется с целью проверки правильности записи на КА плана полета. В предпоследней строке фрагмента задан оператор перехода «УВ», который показывает, что команду РК125 необходимо выдать на КА только в том случае, если будет исполнена команда ПК118. Идентификаторы «ДВ» и «З» в примере означают, что к времени выполнения команды необходимо прибавить длительность ее выдачи, равную 1 с, и задержку выдачи, равную 6 с. Таким образом, общее значение времени команды РК125 составит `(+00:00:09)`.

Операция перехода по неисполнению команд аналогична операции по их исполнению, разница состоит лишь в том, что команда, к которой будет применяться это условие, будет выдаваться по неисполнению предыдущей команды, для которой поставлено данное условие.

Выражения. Выражения состоят из операндов и операций и используются для вычисления значений времени и условия выдачи команды. Примером выражения является запись «`071*(+00:00:02)`» или «`+=3(5)`».

4. **Операторы** реализуют базовые конструкции языка [3]. Для разработки плана полёта используются четыре оператора: выражение, следование, ветвление и цикл. Оператор задаёт законченное описание некоторого действия и состоит из символов, лексем и выражений.

Оператор выражения. Каждое выражение, завершающееся точкой с запятой, рассматривается как оператор, его выполнение заключается в вычислении этого выражения.

Оператор следования. Следованием называется конструкция, представляющая собой последовательное выполнение двух и более операторов. Например, последовательность выполнения команд управления в командных конструкциях задается порядковым номером строки, где размещены команды. Для остальных объектов плана полета последовательность определяется временем их выполнения.

Оператор ветвления задает выполнение либо одного, либо другого оператора в зависимости от выполнения какого-либо условия. Он используется для задания условий выдачи команд управления на КА, передаваемых КИП во время сеанса связи. Существует два типа операторов ветвления: оператор «УВ», который определяет условие выдачи команд, и оператор «УП», который задает условие перехода.

Рассмотрим пример фрагмента плана полета, реализующий повторную запись плана на КА во время проведения сеанса связи. Данная операция во фрагменте выделена серым цветом.

```
3)РК=165(+00:00:02);ДВ=(1);
4)РК=118(+00:00:02);ДВ=(1);
5)РК=118(+00:00:02);ДВ=(1);
  УВ-=1(4);УП-=5(10);З=(5);
6)РК=122(+00:00:02);ДВ=(1);
7)РК=165(+00:00:02);ДВ=(1);
8)РК=125(+00:00:02);ДВ=(1);
9)РК=122(+00:00:02);ДВ=(1);
10)РК=165(+00:00:02);ДВ=(1);
11)РК=113(+00:00:02);ДВ=(1);
12)РК=122(+00:00:02);ДВ=(1);
```

В приведенном примере используется сразу два условных оператора: «УВ» и «УП». Передача плана полета на КА осуществляется командой РК118, она стоит в примере под номером четыре. Процедура передачи и записи плана на КА является непростой операцией и может нести ошибки. Для контроля правильности передачи на КА используется алгоритм подсчета контрольных сумм. Если план полета был передан с ошибкой, то команда, реализующая эту передачу, считается невыполненной и КИП переходит к выдаче следующей команды. Во время выполнения операции по передаче плана может потребоваться резервирование данной операции. Для этого в плане ставится еще одна ко-

манда РК118, к которой применяется условный оператор «УВ». Если первая команда РК118 не была реализована, то КИП начинает выполнять повторную передачу плана полета на КА по второй команде РК118. Если и эта попытка была неудачной, то реализуется операция перехода с помощью условного оператора «УП». Поскольку далее в плане стоит выполнение команды РК125, получение информации оперативного контроля после успешной записи программы полета, то ее выполнение становится нецелесообразным. Переход осуществляется сразу к технологической команде РК165, где далее выполняется операция по сверке бортовой и наземной шкал времени по команде РК113. Если же передача плана полета на КА была успешной, то вторая команда РК118 не реализуется и выдача команд КИП осуществляется в соответствии с планом.

Оператор цикла используется для организации многократно повторяющихся вычислений. Среди командных конструкций плана полета выделяется один класс — циклограммы [4]. Циклограммы предназначены для выполнения на КА. Имеется четыре типа циклограмм: разовые, ежевитковые, суточные и 14-суточные. Разовые циклограммы выполняются на КА только один раз в заданное время, остальные предназначены для многократного выполнения. Для них в плане полета предусмотрена циклическая выдача, она реализуется с помощью указанного оператора «цикла». Для этого после объявления циклограммы необходимо задать тип интервала, используемого для циклической выдачи: 0 — разовая, 1 — ежевитковая, 2 — суточная и 3 — 14-суточная.

Рассмотрим пример:

```
ЦК=ЦГ2"Т-БКУ"(+00:01:05);Ц=(1);
```

В примере приводится циклограмма «ЦГ2»Т-БКУ»», которая выполняет считывание результатов тестирования устройств бортового комплекса управления (БКУ) КА. Это — ежевитковая циклограмма. При составлении плана полета ее необходимо указать один раз, а затем она многократно повторится в плане. Идентификатор «Ц» используется для указания типа цикла повторения. Если повторение циклограммы не задано, то по умолчанию используется значение 0.

Перейдем теперь к рассмотрению более сложных элементов языка разработки плана полета — командных конструкций. Первым элементом в этом классе стоит макрокоманда, обозначаемая идентификатором «МК». Она представляет собой «*структуру*». В языке программирования С++ «структуры» — это массивы, в которых содержатся элементы разных типов. Для языка разработки плана полета структуры могут создаваться только из команд управления, реализуемых на КА. Значение времени выполнения этих команд будет одинаковым, равным времени выполнения самой макрокоманды. Макрокоманды созданы для выдачи на КА многоадресного управляющего воздействия, т.е. для возможности с помощью одной команды управлять несколькими устройствами КА или совершать несколько действий. В состав макрокоманд, кроме программных команд, входят командные слова, слова данных, ответные слова, информационные слова, служебные слова и другие управляющие структуры, составляющие множество команд управления КА [4].

Сделаем описание макрокоманды:

имя(*значение*){
  1)*тип_команды=имя1*;
  2)*тип_команды=имя2*;
  ...
  N)*тип_команды=имяN*;
}

Каждая макрокоманда имеет имя и тело, заключенное в фигурные скобки. В теле макрокоманды команды управления перечисляются в виде нумерованного списка. Порядок выполнения команд определяется последовательностью их выдачи бортовым комплексом управления, который отвечает за реализацию команд на КА.

Рассмотрим простой пример макрокоманды «КБШВ», у которой все команды, входящие в нее, имеют один тип. Данная макрокоманда входит в состав циклограммы «ЦГ4»БШВ-КНА»», обеспечивающей передачу в систему сбора и регистрации научной информации текущего кода бортовой шкалы времени:

```
ЦГ4"БШВ-КНА"(+00:00:00){
  Т0=(+00:00:00)
  1)МКО=РСВ*(+00:00:00);
  2)МК=КБШВ*(+00:00:01){
       1)МКО=ЧТР3;
       2)МКО=ЧТР2;
       3)МКО=ПБШВ;
  };
}
```

Первой командой в циклограмме стоит команда, обеспечивающая подготовку кода текущего времени устройством КА, называемым бортовым стандартом времени и частоты (БСВЧ). Идентификатор «МКО» означает, что команда является программной командой, передаваемой по мультиплексному каналу обмена КА. Второй командой в циклограмме стоит макрокоманда, содержащая три команды. Первая команда обеспечивает чтение кода

часов и суток из БСВЧ, вторая — чтение кода секунд и минут, а третья осуществляет передачу кода. Времена выполнения всех трех команд заданы как (+00:00:01) и отсчитываются от времени выполнения первой команды циклограммы.

В приведенном примере тело макрокоманды описано непосредственно в теле циклограммы. Для упрощения вычислений все тела макрокоманд были перенесены в таблицу БД программного комплекса автоматизированного планирования полёта, поэтому перед использованием макрокоманда, так же, как и команды управления, должна быть описана в БД. Идентификатор «МК» указывает, в какой таблице БД содержится данная макрокоманда, а имя — в какой строке ее следует искать.

Следующим рассматриваемым элементом является циклограмма, обозначаемая идентификатором «ЦК». Циклограмма представляет особый тип данных под названием «класс». К этому же типу относятся и другие командные конструкции: операции управления «О» и режимные операции «Р». Для языка разработки плана полета, как и для языка программирования C++, «класс» — это абстрактный тип данных, который представляет собой модель выполнения какой-либо операции, совершаемой КА и НАКУ. Элементами «класса» могут быть команды управления, макрокоманды и другие классы, кроме данного.

Приведем описание структуры класса:

имя(*значение*){
  Т0=(значение);
  1)ПК | МК=*имя1*(*значение*);[ГИ;]
  2)РК=*имя2*(*значение*);[ДВ; З; ОВ; УВ; УП;]
  3)ЦК | О | Р=*имя3*(*значение1, значение2, ..., значениеN*);[ГИ;]
  ...
  N)*тип_данных*=*имяN*(*значение*);
}

Так же, как и «структура», «класс» имеет имя и тело. Все элементы в «классе» перечисляются в виде нумерованного списка, за исключением первого. Первым элементом стоит виртуальная команда. У нее нет собственного имени, она задает значение базового времени из плана пространственно-временных действий КА. Дальнейшее расположение элементов класса может быть любым. Порядок выполнения команд управления и командных конструкций класса определяется их последовательностью в списке.

В качестве примера рассмотрим «класс», моделирующий операцию получения телеметрической информации с КА по циклограмме «ЦГ13"Т3"», которая обеспечивает воспроизведение всех записывающих устройств бортовой аппаратуры телесигнализации (БАТС) с последующим включением режима записи:

```
ЦГ13"Т3"(+00:08:11){
  Т0=(+00:00:00);
  1)ПК=017*(+00:00:00);
  2)ПК=037*(+00:00:20);
  3)ПК=038*(+00:02:20);
  4)ПК=028*(+00:02:30);
  5)ПК=039*(+00:02:31);
  6)ПК=040*(+00:02:40);
  7)БАТС=ЗапПЗ*(+00:07:05);
  8)ПК=039*(+00:07:10);
  9)ПК=035(+00:07:12);
  10)ПК=038*(+00:07:20);
  11)ПК=037*(+00:07:30);
  12)ПК=042(+00:07:32);
  13)ПК=028*(+00:10:54);
  14)ПК=016*(+00:10:56);
}
```

В приведенном примере содержится виртуальная команда, определяющая базовое время отсчета выполнения команд управления «Т0» и программные команды управления. Первая программная команда обеспечивает включение телеметрического передатчика КА, следующая команда включает режим передачи информации, далее по командам выполняется воспроизведение информации из запоминающих устройств КА, затем следует выключение режима воспроизведения, перевод запоминающих устройств КА в исходное состояние, включение режима записи информации и отключение передатчика. У двух команд в этой циклограмме отсутствует признак «*». Это говорит о том, что циклограмма изначально предназначена для использования без включения режима записи. Для того чтобы включить режим записи в циклограмме, необходимо задать выполнение этих двух команд. Выполнение команд задается в заголовке циклограммы:

```
ЦК=ЦГ13"Т3"(+00:08:11,035,042);
```

У каждого «класса» командных конструкций существует инициализирующая его функция. Она вызывается автоматически при создании объекта «класса». В функции передаются базовое значение времени для выполнения расчетов и «мягкие» команды управления, необходимые для их включения в план. Если все команды циклограммы «жесткие», то в функции передается только значение базового времени.

Рассмотрим еще особенности задания в «классе» некоторых его элементов. Для всех команд уп-

равления, выдаваемых КА, и командных конструкций предусмотрен механизм, позволяющий сделать автоподстановку значения времени, если время выдачи не определено. Например, продолжительность сеанса связи с КА на каждом витке будет разная, соответственно, время сеанса должно меняться в зависимости от витка. Задание времени осуществляется через элемент «ГИ». Для этого в «классе» после задания команды необходимо сослаться на элемент плана полета, в котором будет находиться время соответствующего интервала. В плане полета на каждый «класс» предусмотрено десять таких элементов для десяти интервалов: «ГИ=ГИ0, ГИ1, ..., ГИ9». В самом плане полета после задания элемента необходимо указать время, передаваемое в «класс»:

```
ЦК=ЦГ44"ВКС"(+00:01:05);ГИ0=(+00:01:48);
```

Тогда в теле «класса» по ссылке «ГИ0» передастся значение времени, которое необходимо добавить к значению времени выдаваемой команды:

```
Т0=(+00:00:00);
1)ПК=018*(+00:00:00);
2)ПК=019*(+00:10:00);ГИ=ГИ0;
```

Таким образом, для циклограммы «ЦГ44"ВКС"», обеспечивающей организацию сеанса связи с КА, время сеанса увеличится на 1 мин 48 с по сравнению с исходным, равным 10 мин.

Существенным свойством «класса» является то, что детали его реализации скрыты в теле «класса». Поскольку так же, как и для «структур», все тела классов командных конструкций помещены в БД программного комплекса автоматизированного планирования полёта, человек, составляющий план полета, оперирует только заголовками «класса». Таким образом, «класс», рассматриваемый как модель операции, становится «черным ящиком», замкнутым по отношению к другим операциям. Использование таких «черных ящиков» при составлении плана полета существенно облегчает работу специалистов группы управления, повышает надежность и качество составления плана.

В языке программирования C++ широко используется механизм наследования «классов» [3]. Он позволяет строить иерархии «классов», в которых производные «классы» получают элементы родительских «классов» и могут дополнять или изменять их свойства. Наследование позволяет объединять общие свойства для нескольких «классов» в одном, используя его в качестве базового класса. «Классы», находящиеся ближе к началу иерархии, объединяют в себе наиболее общие черты для всех нижележащих «классов». По мере продвижения вниз по иерархии «классы» приобретают все больше конкретных черт.

Механизм наследования «классов» используется и для языка разработки плана полета. Рассмотренный выше «класс» циклограмм является базовым и стоит на первой ступени иерархии, производным от него является «класс» операции управления. А от «класса» операции управления происходит «класс» режимных операций. При наследовании «классов» основным вопросом является организация доступа к элементам родительского «класса». В языке разработки плана полета доступ к переменным родительского «класса» осуществляется через представителя этого «класса».

Рассмотрим подробнее производные «классы».

«Класс» операций управления предназначен для группировки отдельно стоящих циклограмм в операции управления, выполняемые на КА. Элементами этого «класса» могут быть команды управления, макрокоманды и циклограммы. В отличие от «класса» циклограмм, который может оперировать только командами управления и макрокомандами, «класс» операций управления может строить более сложные конструкции. Рассмотрим пример:

```
ОУ"СС"(+00:08:48){
  Т0=(+00:00:00);
  1)ЦК=ЦГ44"ВКС"(-00:05:38);
    ГИ0=(+00:04:26);
  2)ЦК=ЦГ13"Т3"(+00:00:22,035,042);
  3)ПК=023*(+00:10:51);
}
```

Операция управления «ОУ"СС"» осуществляет подготовку и проведение сеанса связи с КА. Она обеспечивает включение и выключение передатчика командной радиолинии по циклограмме «ЦГ44"ВКС"», выполняет организацию передачи телеметрической информации по циклограмме «ЦГ13"Т3"» с последующим включением режима записи, выполняет обнуление счетчика аварийно-временного устройства по программной команде ПК023. Доступ к командам циклограммы осуществляется через «класс» циклограмм.

Приведем еще пример операции управления:

```
ОУ"Т-БКУ/ССРНИ"(+00:01:00){
  ЦК=ЦГ3"Т-ССРНИ"(+00:00:00);
  ЦК=ЦГ2"Т-БКУ"(+00:00:05);
}
```

Данная операция содержит две циклограммы, выполняющие считывание результатов тестирования устройств системы сбора и регистрации научной информации и бортового комплекса управления.

У «класса» операций управления существует еще одна особенность: с его помощью можно создавать групповые операции. Групповая операция представляет собой объединение нескольких операций под одним именем. По своим свойствам она схожа со «структурой», рассматриваемой для макрокоманд. Приведем пример групповой операции:

```
ГОУ"ЗОгр."(+00:13:49){
  О=ОУ"З1"(+00:00:00);
  О=ОУ"З2"(+00:18:00);
  О=ОУ"З3"(+00:38:00);
  О=ОУ"З4"(+01:00:00);
}
```

Групповая операция «ГОУ"ЗОгр."» используется для задания зон ограничений работы научной аппаратуры КА. В нее входят четыре операции, соответствующие четырем зонам ограничений. В каждой операции содержится по две циклограммы: первая выполняется при входе КА в зону ограничения, вторая — при выходе:

```
ОУ"З1"(+00:13:49){
  Т0=(+00:00:00);
  ЦК=ЦГ32"НЗ-1"(+00:00:00);
  ЦК=ЦГ36"КЗ-1"(+00:05:10);
}
```

«Класс» режимных операций является производным от «класса» операций управления. Он предназначен для создания операций управления, используемых в сеансе связи с КА. Рассмотрим пример, в котором реализуется передача плана полета на КА с резервированием данной операции:

```
Р"ЗапРП3"(+00:08:48){
  Т0=(+00:00:00);
  1)О=ОУ"СС"(+00:00:00);
  2)РК=165(+00:00:02);ДВ=(1);
  3)РК=118(+00:00:02);ДВ=(1);
  5)РК=118(+00:00:02);ДВ=(1);
    УВ-=1(3);З=(6);
  4)РК=122(+00:00:02);ДВ=(1);
}
```

В примере первой стоит операция управления «ОУ»СС»», обеспечивающая подготовку и проведение сеанса связи, далее следуют разовые команды управления, по которым осуществляется передача плана полета на КА. Команда РК118 является режимной, по ней на КА включается режим записи. Отключение режима производится по команде РК122. Для второй команды РК118 здесь задан оператор «УВ», с помощью которого, в случае неудачной первой попытки, выполняется повторная передача плана по резервной команде.

Перейдем теперь к рассмотрению процедуры построения плана полета с использованием языка разработки плана полета. Приведем фрагмент плана полета:

```
ПП=(ДН=25.06.2009,ДК=09.07.2009){
  Д=(25.06.2009){
      В=(2197){
          ЦК=ЦГ4"БШВ-КНА"(+00:00:00);Ц=(1);
          О=ОУ"Т-БКУ/ССРНИ"(+00:01:00);Ц=(1);
          ЦК=ЦГ44"ВКС"(+00:01:09);ГИ0=(+00:01:48);
          ЦК=ЦГ5"СВЕТ"(+00:04:01);
          ЦК=ЦГ13"Т3"(+00:08:11,035,042);
          КИП=(33);НОС=(+00:01:15);ТП=(2);
      СР=(с16,с84р01р03р14р16р17,с96р12р61);
          {
              1)РК=153;ДВ=(1);
              3)Р=Р"ЗапРП1"(+00:00:09);
              4)ТЕКСТ=(С3_02197_01_1);
              5)Р=Р"ЗапРП1"(+00:00:15);
              6)ТЕКСТ=(С3_02197_02_1);
              7)Р=Р"ИОК2"(+00:00:11);
              8)Р=Р"СВ"(+00:00:09);
              9)Р=Р"ИОК1"(+00:00:09);
              10)РК=159;ДВ=(1);Ф=(1);Т=(16:04:30);
          }
          О=ГОУ"ЗОгр."(+00:13:49);
          ЦК=ЦГ6"ТЕНЬ"(+01:15:15);
      }
```

```
        В=(2198){
             ЦК=ЦГ44"ВКС"(+00:03:10);ГИ0=(+00:01:30);
             ЦК=ЦГ5"СВЕТ"(+00:04:13);
             КИП=(35);НОС=(+00:01:20);ТП=(2);
        СР=(с16,с84р01р03р14р17);
             {
                  1)РК=153;ДВ=(1);
                  2)Р=Р"ЗапРП1"(+00:00:09);
                  3)ТЕКСТ=(С3_02198_01_1);
                  4)Р=Р"ИОК2"(+00:00:14);
                  5)Р=Р"ИОК1"(+00:00:09);
                  6)РК=159;ДВ=(1);Ф=(1);Т=(17:41:09);
             }
             О=ГОУ"ЗОгр."(+00:14:00);
             ПК=308(+00:22:18);
             ПК=316(+00:22:20);
             ...
        }
        В=(2199){
             ...
   }
        Д=(26.06.2009){
        ...
}
```

Рассмотрен фрагмент плана полета, в котором на соседних двух витках организовано проведение сеансов связи с КА. Составление плана полета начинается с задания интервала, на котором реализуется план. Для этого используется идентификатор «ПП», у которого в значении указываются даты начала «ДН=25.06.2009» и конца «ДК=09.07.2009» интервала планирования. Затем выполняется открытие блока, в котором будет находиться данный интервал. После этого создается блок на суточный интервал. Идентификатор «Д» используется для задания даты «Д=(25.06.2009)» суточного интервала. Далее суточный интервал делится на витковые интервалы, которые задаются с помощью идентификатора «В». На витковом интервале выполняется расстановка команд управления, макрокоманд, циклограмм, операций и задание интервалов зон проведения сеансов связи; в самих зонах сеансов используются только разовые команды и режимные операции управления. Проведение сеансов реализуется наземными командно-измерительными станциями. Для задания интервала их работы используется идентификатор «КИП», в котором указывается номер задействуемой станции «КИП=(33)». По номеру станции из плана пространственно-временных действий выбираются данные, определяющие местоположение станции и время ее работы. Идентификаторы «НОС», «КОС» и «НОС/КОС» позволяют изменять время начала и окончания сеанса связи с КА. В приведенном примере время начала сеанса на витке 2197 задержано на 1 мин 15 с. Идентификатор «ТП» указывает, что выдачу разовых команд управления необходимо выполнять с интервалом в 2 с, он используется для упрощения задания времени выполнения большого числа команд. Идентификатор «СР» задает список задействуемых средств станции и режимов работы этих средств, перечисляемых через «,». Данные средства используются для проведения сеанса связи. Разовые команды управления и режимные операции, реализуемые в сеансе связи, расставляются в порядке их передачи на КА. Каждый элемент, в том числе и комментарий, используемый для составления сеанса связи, должен быть пронумерован. Идентификаторы «Ф» и «Т», используемые для разовых команд в конце сеанса, позволяют устанавливать фиксированное время выдачи команд. Соответственно «Ф=(1)» означает, что для команды задан признак ее фиксации, а «Т=(16:04:30)» — что команда будет выдана в 16 час 4 мин 30 с.

Выводы

1. Язык разработки плана полета был создан для описания командных конструкций и подготовки исходных данных плана полёта, составляемого автоматизированным способом на программном ком-

плексе автоматизированного планирования полёта автоматических КА. Его основой стали языки программирования высокого уровня C++ и C#.

2. Созданный язык разработки плана полета позволил структурировать исходные данные, сделать их легко читаемыми. Основными элементами плана полета являются команды управления. Упрощенная структура описания плана полета позволила представить план в виде отдельных блоков и скрыть от человека детали его реализации с помощью «классов», используемых для описания командных конструкций, в основе которых стоят команды управления.

3. Применение языка разработки плана полета в программном комплексе планирования полета автоматических КА позволило сократить количество «операторов», используемых для описания исходных данных. Благодаря этому удалось уменьшить время, отводимое на подготовку исходных данных, и количество ошибок, вносимых в план.

4. Язык разработки плана полета в составе программного комплекса планирования полета был внедрен в ЦУП и использовался для подготовки исходных данных при управлении полётом автоматических КА различного назначения.

Библиографический список

1. *Жигастова О.К., Почукаев В.Н.* Программный комплекс автоматизированного планирования полёта автоматических космических аппаратов // Вестник Московского авиационного института. 2014. Т. 21. №4. С. 60-70.
2. *Жигастова О.К., Почукаев В.Н.* Ключевые операци системы управления полётами автоматических околоземных космических аппаратов // Труды МАИ [Электронный ресурс]. 2011. №49. Режим доступа: http://www.mai.ru/science/trudy/published.php?ID=28296
3. *Павловская Т.А.* C/C++. Программирование на языке высокого уровня. — СПб.: Питер, 2002. — 464 с.
4. *Жигастова О.К., Почукаев В.Н.* Командные конструкции и использование их при автоматизированном планировании полёта автоматических космических аппаратов // Вестник Московского авиационного института. 2012. Т. 19. №5. С. 21-31.
5. *Албахари Джозеф, Албахари Бен.* C#5.0. Справочник. Полное описание языка / Пер. с англ. Ю.Н. Артеменко. — 5-е изд. — М.: Вильямс, 2013. — 1008 с.

FLIGHT PLAN DEVELOPMENT LANGUAGE ALLOWING AUTOMATIC FLIGHT PLANNING FOR AUTOMATED SPACECRAFT

Zhigastova O.K.*, Pochukayev V.N.

Central Research Institute of Machine Building,
4, Pionerskaya str., Korolev, Moscow region, 141070, Russia
** e-mail: inolga-ok@yandex.ru*

Abstract

Purpose

The article considers the flight plan development language, used in software complex for automated spacecraft (SC) flight planning [1] to describe instruction structures [4] and prepare the initial flight plan data.

Design/methodology/approach

The language of the flight plan development is based on methods of structural, modular and object-oriented programming. [3] Like other high-level languages, the language of the flight plan development consists of characters, tokens (basic structure), expressions, operators and classes.

The problem of developing a new language stems from the need to develop a tool allowing describe commands used to plan the flight control and actions performed according to them.

The flight plan development language uses six types of data such as integer, character, logic, calendar date, time and n-tuple to store control commands parameters vector.

All calculations made are control commands execution time operations. This language defines variables for command time values storage. Each variable consists of type and name. The type determines the properties of a variable, and the name points for which command it is necessary to make calculation. Calculation of values is performed using the operations, which determine what time operations should be performed.

In addition to time operations, this language defines execution of conditional operations. Conditional

operations are used to set the conditions for issuing control commands implemented during n a spacecraft communication with ground control station.

To provide repetitive calculations the language uses the cycle statement.

To describe more complex language constructions, used to compile a flight plan, «classes» were included. These "Classes" are used to describe the command structures. They represent a model carrying out an action executed by either SC, or a Ground Automated Control Complex. The elements that constitute such a structure can be both control commands and command structures themselves except the given one.

The language of the flight plan development was created for a formalized description of the command structures and the initial data used to compile a flight plan in an automated way. It allows describe the flight plan elements and structures more clearly, hiding the details of their realization. Its helps to provide the input data preparation for an automated spacecraft flight planning software complex.

Findings

This work results in creation of the flight plan development language for an unmanned spacecraft automated flight planning software complex. It aid in the description of structures and preparation of the input data used for compiling the flight plan in an automated way.

The developed language made it possible to formalize the description of the flight plan command structures, the initial data structure and simplify the way to describe them, while reducing the number of "operators" used in the description.

The created flight plan development language was implemented at the Mission Control Center (MCC) software as part of a complex of automatic spacecraft mission planning and has been used to prepare the initial data for the flight control of space crafts of scientific and socio-economic purpose.

Research limitations/implications

The flight plan development language can be applied during preparation of the flight plan initial data to control an automatic spacecraft of scientific and socio-economic purpose.

Originality/value

1. The flight plan development language has been created to describe the command structures and initial data preparation for flight plan made by an automated way with use of software complex of unmanned spacecraft automated flight planning. It was designed based on high level programming languages C ++ and C #.

2. The created flight plan development language allows structuring the initial data and to make it easy to read. Control commands are the basic elements of the flight plan. A simplified description of the flight plan structure allowed presenting a plan in the form of separate blocks and hiding from a human the details of its implementation using "classes" implemented to describe the command structures, which are based on the control commands.

3. The application of the flight plan development language for flight planning software complex of automated spacecraft has reduced the number of "operators" necessary to describe the initial data. This allowed reducing the time assigned for the preparation of the initial data and the number of errors introduced into the plan.

4. The flight plan development language as a part of flight planning software system was implemented in the MCC and was used for preparation of the initial data for an unmanned spacecraft of scientific and socio-economic purpose flight control.

Keywords: automatic spacecraft, flight planning, flight plan, command structure, flight planning language, method of flight planning, automation of planning.

References

1. Zhigastova O.K., Pochukayev V.N. *Vestnik Moskovskogo aviatsionnogo instituta*, 2014, vol. 21, no. 4, pp. 60-70.
2. Zhigastova O.K., Pochukayev V.N. *Trudy MAI*, 2011, no. 49, available at: http://www.mai.ru/science/trudy/eng/published.php?ID =28296 (accessed 27.12.2011).
3. Pavlovskaya T.A. *C/C++. Programmirovanie na yazyke vysokogo urovnya (C/C++. High-level language Programing)*, St. Petersburg, Piter, 2002, 464 p.
4. Zhigastova O.K., Pochukayev V.N. *Vestnik Moskovskogo aviatsionnogo instituta*, 2012, vol. 19, no. 5, pp. 21-31.
5. Alahari Joseph, Alahari Ben. *C#5.0. Spravochnik. Polnoe opisanie yazyka* (*C#5.0. in a Nutshell: The Definitive Reference, Fifth Edition*), Moscow, Vil'yams, 2013, 1008 p.