

## **ВВЕДЕНИЕ**

Цель – составить программу для создания базы данных запуска ракет, в которую вначале вводится количество запусков, потом по каждому полёту дату запуска, тип ракеты-носителя, полезную нагрузку, заказчика, космодром, результат запуска, после чего программа даёт пользователю выбор параметра, по которому можно отфильтровать запуск, после чего пользователь вводит этот параметр и ему выдаётся информация об определённом полёте.

Задачи:

- Составить IDEF0-диаграмму проекта;
- Составить блок-схемы алгоритмов;
- Реализовать алгоритмы на языке Object Pascal;
- Протестировать реализации алгоритмов.

## Аналитическая часть

Структура представлена на рисунке 1 в виде IDEF0-нотации. Задача разделяется на несколько блоков, представленных на рисунке 2:

- Выполнение ввода данных по количеству запусков, информации по каждому запуску;
- Запись введённых данных в файл;
- Поиск по определённому параметру информации о запуске.
- Вывод данных на экран.

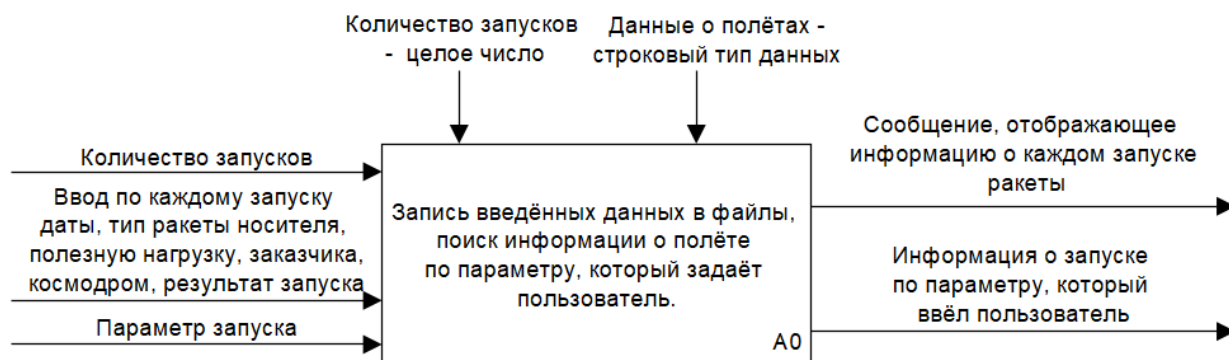


Рисунок 1 – Общая IDEF0-нотация

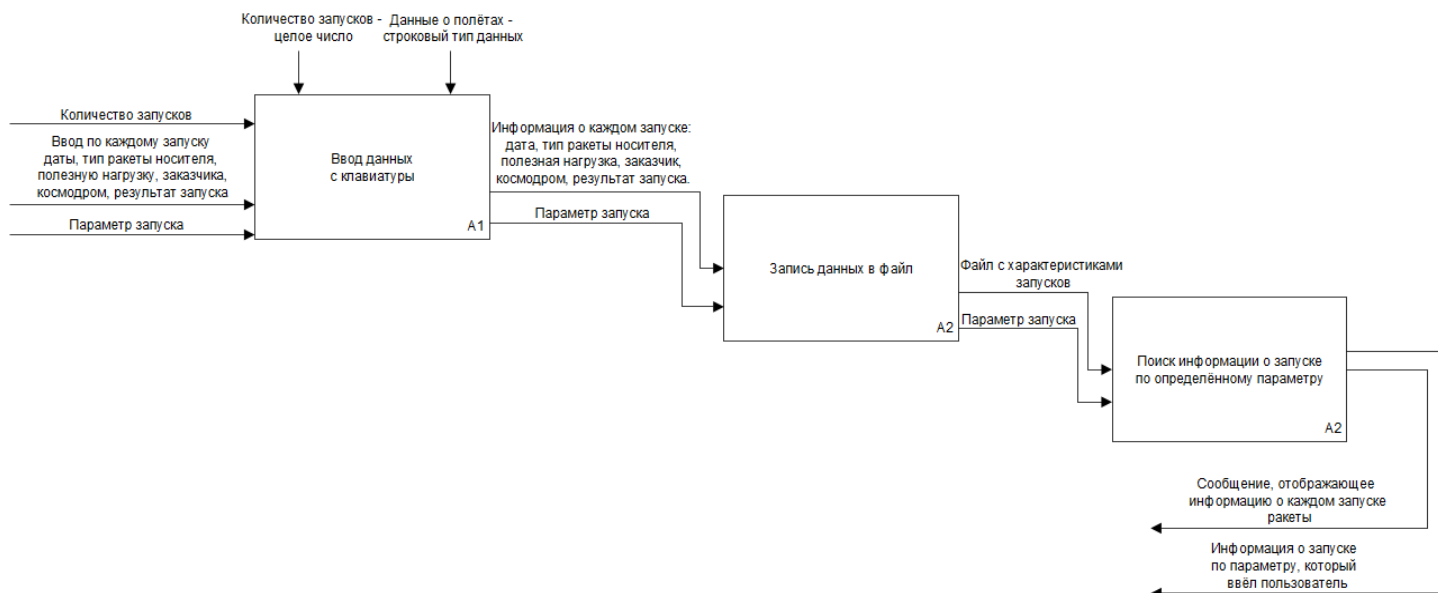


Рисунок 2 – Подробная IDEF0-нотация

## Конструкторская часть

На основании изученных источников была составлена следующая структура записи информации о запуске ракеты:

Таблица 1 — структура записи «Данные о запусках».

Поле записи	Тип данных
Дата запуска	строка
Тип ракеты - носителя	строка
Полезная нагрузка	строка
Заказчик	строка
Космодром	строка
Результат запуска	строка

Блоки IDEF0-диаграммы представляют собой 1-2 действия, ввиду чего рациональнее отобразить алгоритм всего проекта целиком без разбиения каждого блока на отдельные процедуры.

Блок-схема алгоритма представлена на рисунках 1.1 и 1.2. Блок-схема алгоритма процедуры VvodZapusk представлена на рисунках 2.1 и 2.2. Блок-схема алгоритма процедуры VvodZapuskov представлена на рисунке 3. Блок — схема алгоритма процедуры VypisZapuskov представлена на рисунке 4. Блок — схема алгоритма процедуры PoiskZapuskovPoKriteriyam представлена на рисунке 5. Блок — схема алгоритма процедуры Zagruzkaizfaily представлена на рисунках 6.1 и 6.2. Блок — схема алгоритма процедуры SohranenieVFaile представлена на рисунке 7. Элементы, отвечающие за интерфейс пользователя, на блок-схемах не отображены; текстовые сообщения, ввиду малозначимости их дословного приведения, представлены сокращенно.

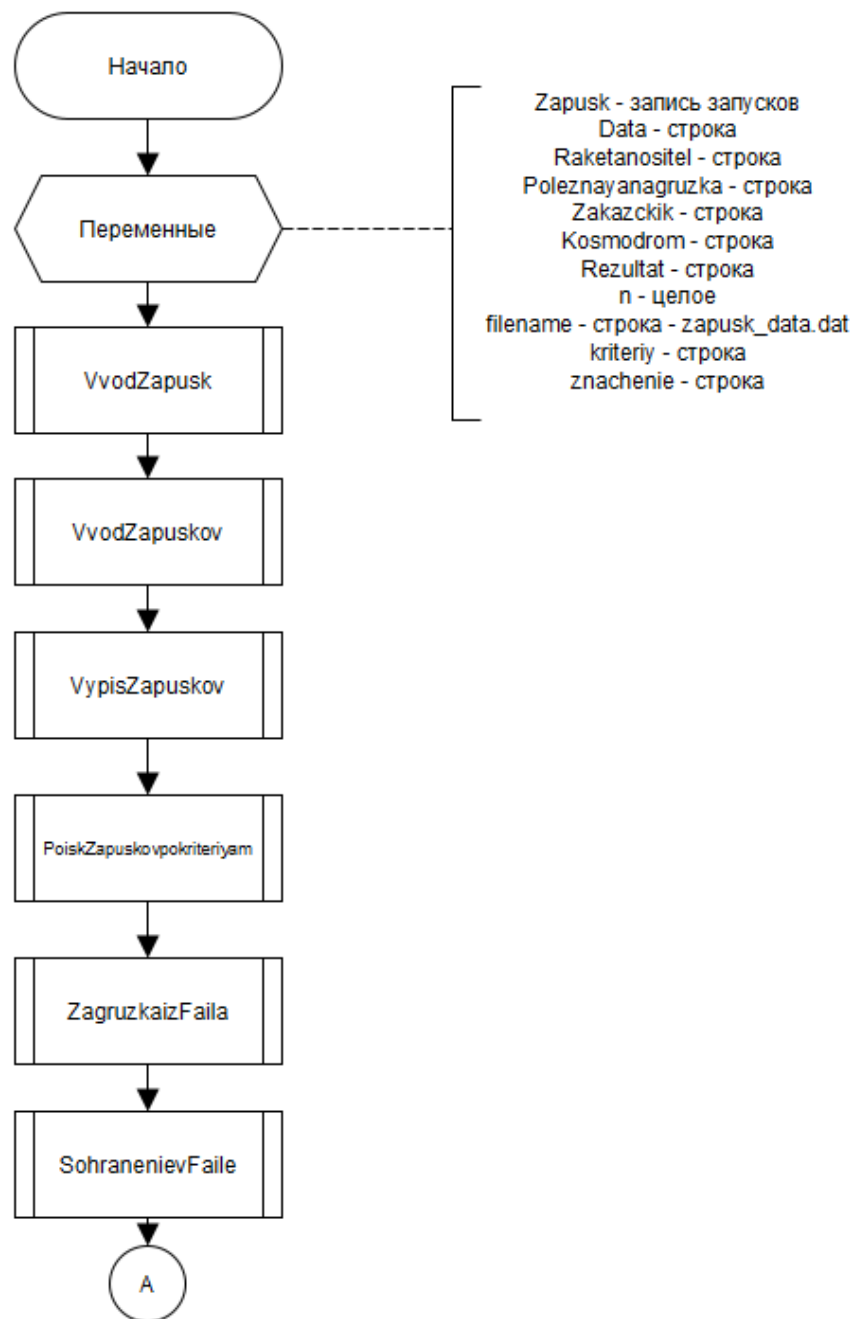


Рисунок 1.1 – Блок-схема алгоритма программы

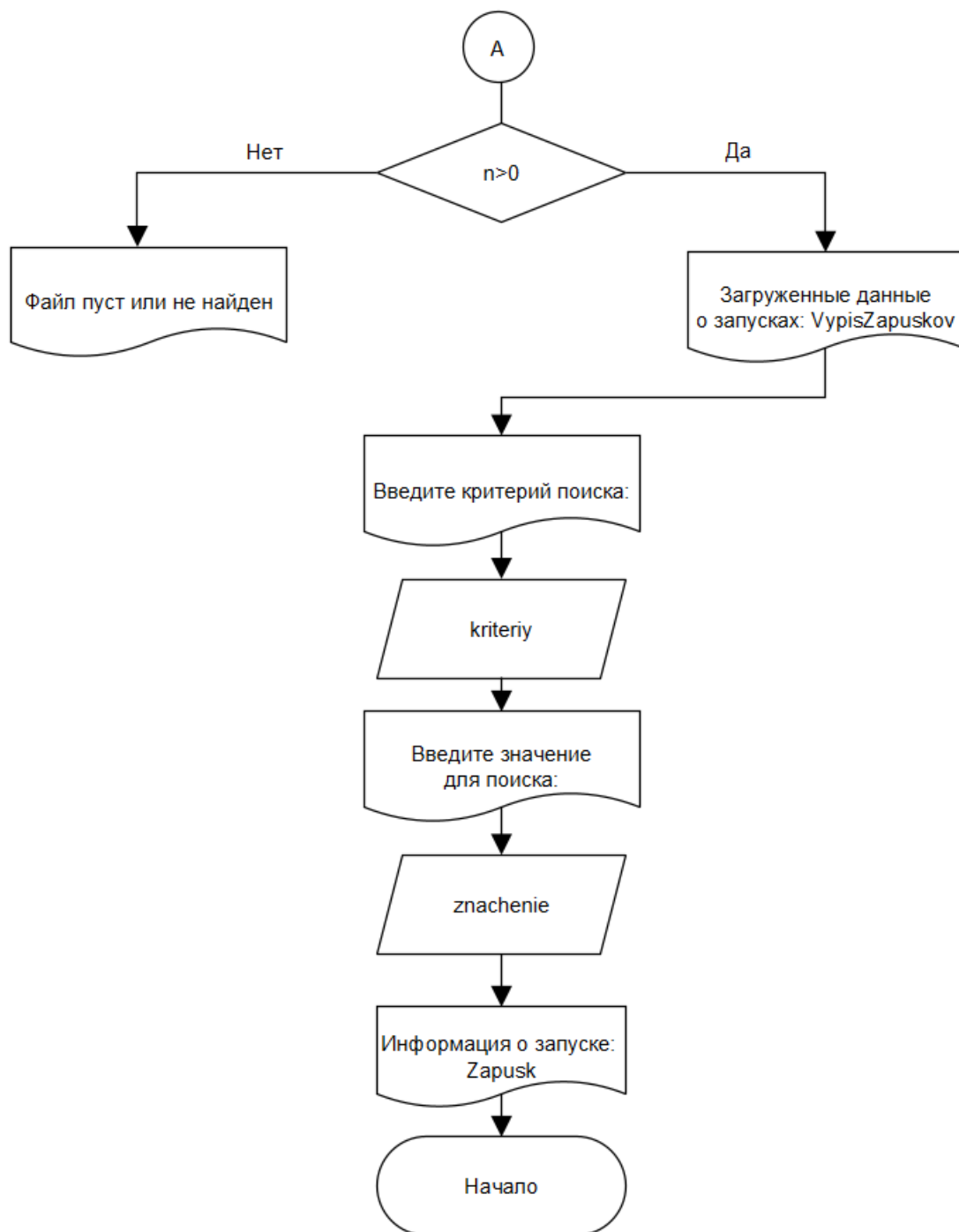


Рисунок 1.2 — Блок — схема алгоритма программы



Рисунок 2.1 – Блок — схема алгоритма процедуры VvodZapusk

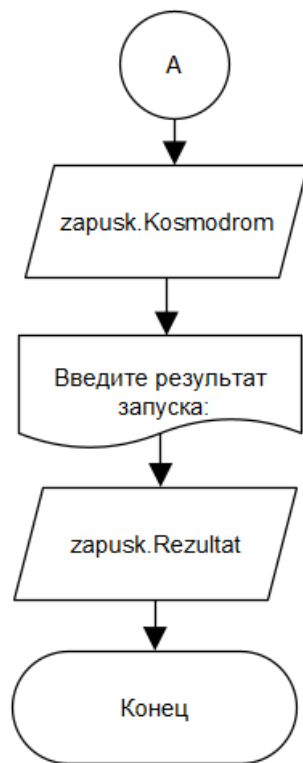


Рисунок 2.2 — Блок — схема алгоритма процедуры VvodZapusk

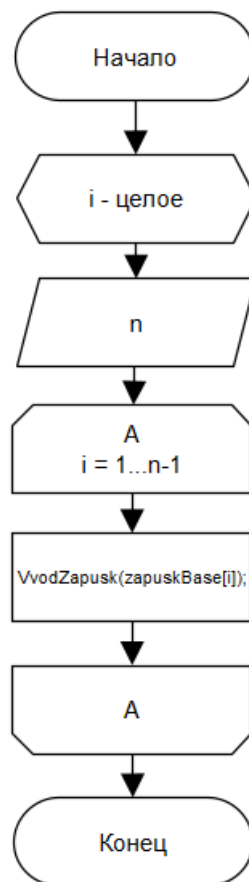


Рисунок 3 — Блок — схема алгоритма процедуры VvodZapuskov

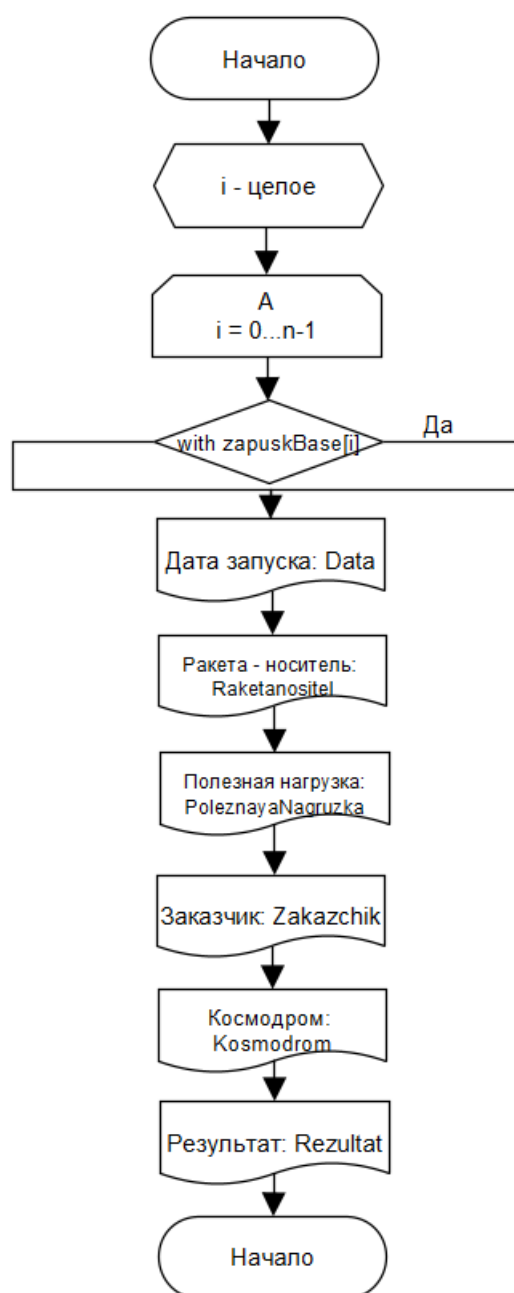


Рисунок 4 — Блок — схема алгоритма процедуры VypisZapuskov



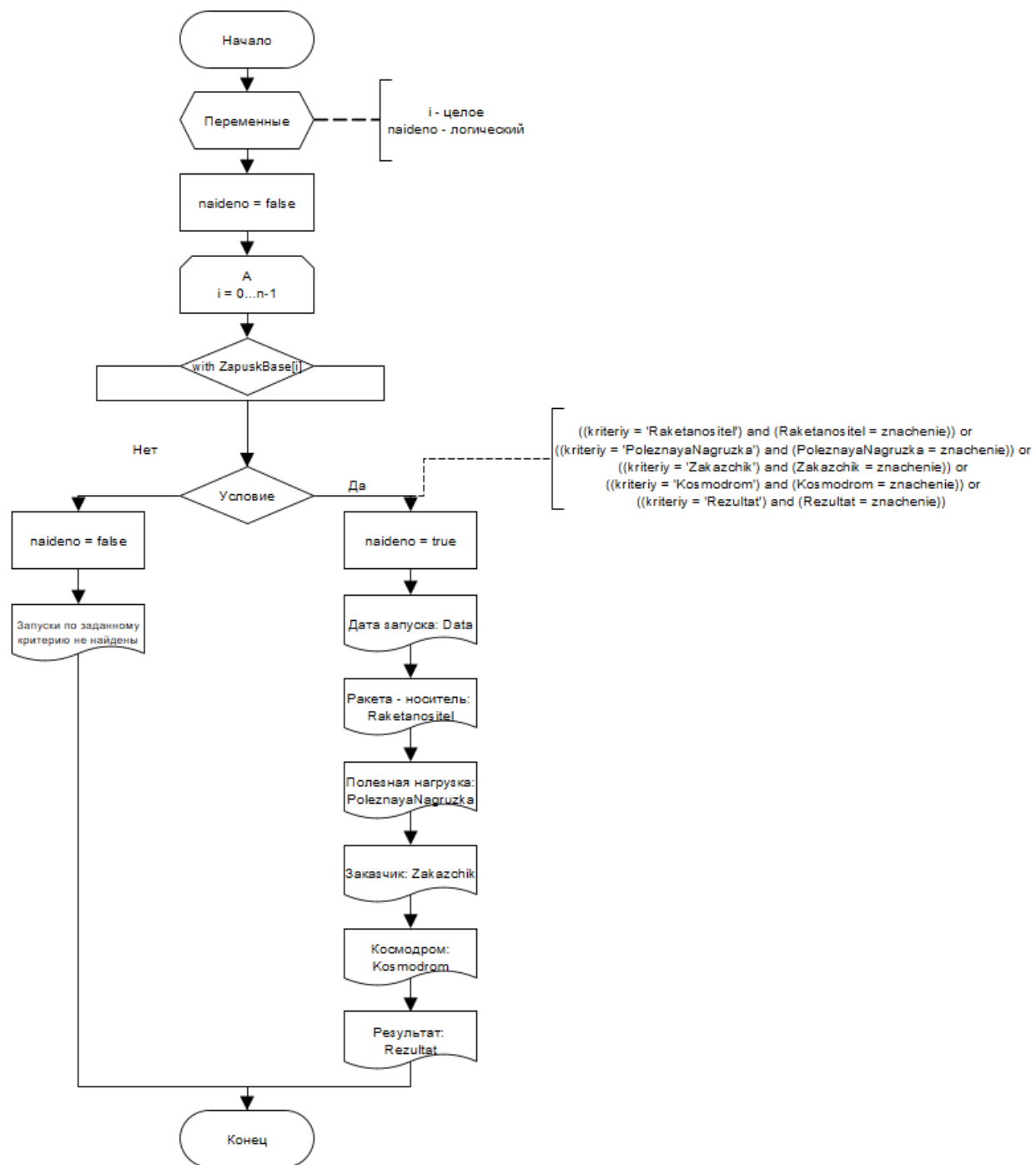


Рисунок 5 — Блок — схема алгоритма процедуры PoiskZapuskovPoKriteriyam

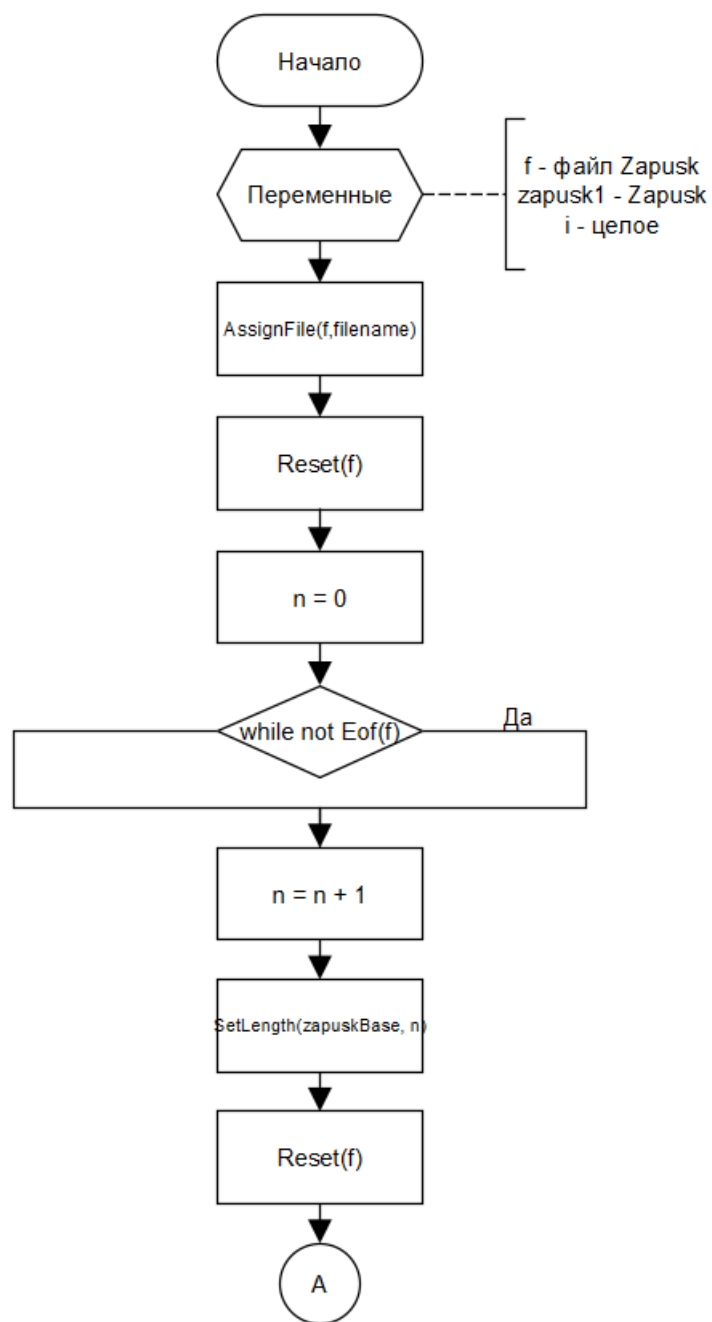


Рисунок 6.1 — Блок — схема алгоритма процедуры *Zagruzkaizfaiila*

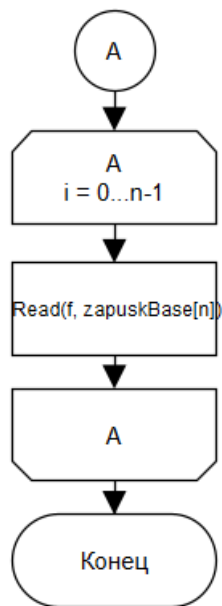


Рисунок 6.2 — Блок — схема алгоритма процедуры Zagruzkaizfaile

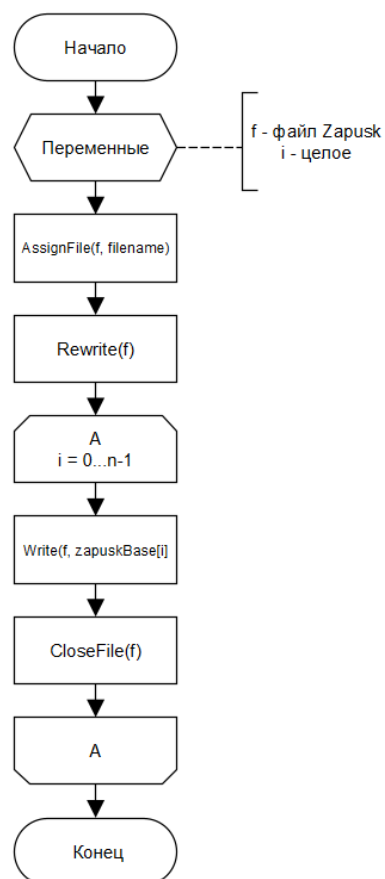


Рисунок 7 — Блок — схема алгоритма процедуры SohranenieVFaile

## Технологическая часть

### Реализация алгоритма

В настоящем разделе представлена реализация алгоритма, чья блок-схема представлена на рисунке 1. Реализация была произведена с помощью языка программирования Pascal и представлена в листинге 1.

Листинг 1 – Программа алгоритма

```
program lab5;
```

```
Type
```

```
  Zapusk = record
```

```
    Data: string[20];
```

```
    Raketanositel: string[20];
```

```
    PoleznayaNagruzka: string[20];
```

```
    Zakazchik: string[20];
```

```
    Kosmodrom: string[20];
```

```
    Rezultat: string[20];
```

```
end;
```

```
  ZapuskBase = array of Zapusk;
```

```
procedure VvodZapusk(var zapusk: Zapusk);
```

```
begin
```

```
  writeln('Введите дату запуска:');
```

```
  readln(zapusk.Data);
```

```
  writeln('Введите тип ракеты-носителя:');
```

```
  readln(zapusk.Raketanositel);
```

```
  writeln('Введите полезную нагрузку:');
```

```
  readln(zapusk.PoleznayaNagruzka);
```

```
  writeln('Введите заказчика:');
```

```
  readln(zapusk.Zakazchik);
```

```
  writeln('Введите космодром:');
```

```
  readln(zapusk.Kosmodrom);
```

```
  writeln('Введите результат запуска:');
```

```
  readln(zapusk.Rezultat);
```

```
end;
```

```
procedure VvodZapuskov(var zapuskBase: ZapuskBase; var n: integer);
```

```
var
```

```
  i: integer;
```

```
begin
```

```
  writeln('Введите количество запусков:');
```

```
  readln(n);
```

```
  SetLength(zapuskBase, n);
```

```
  for i := 0 to n - 1 do
```

```
    VvodZapusk(zapuskBase[i]);
```

```
end;
```

```
procedure VypisZapuskov(zapuskBase: ZapuskBase; n: integer);
```

```

var
  i: integer;
begin
  for i := 0 to n - 1 do
    with zapuskBase[i] do
      begin
        writeln('Дата запуска: ', Data);
        writeln('Ракета-носитель: ', Raketanositel);
        writeln('Полезная нагрузка: ', PoleznayaNagruzka);
        writeln('Заказчик: ', Zakazchik);
        writeln('Космодром: ', Kosmodrom);
        writeln('Результат: ', Rezultat);
        writeln;
      end;
    end;
end;

procedure PoiskZapuskovPoKriteriyam(zapuskBase: ZapuskBase; n: integer; kriteriy: string; znachenie: string);
var
  i: integer;
  naideno: boolean;
begin
  naideno := false;
  for i := 0 to n - 1 do
    with zapuskBase[i] do
      if ((kriteriy = 'Raketanositel') and (Raketanositel = znachenie)) or
        ((kriteriy = 'PoleznayaNagruzka') and (PoleznayaNagruzka = znachenie)) or
        ((kriteriy = 'Zakazchik') and (Zakazchik = znachenie)) or
        ((kriteriy = 'Kosmodrom') and (Kosmodrom = znachenie)) or
        ((kriteriy = 'Rezultat') and (Rezultat = znachenie)) then
        begin
          writeln('Дата запуска: ', Data);
          writeln('Ракета-носитель: ', Raketanositel);
          writeln('Полезная нагрузка: ', PoleznayaNagruzka);
          writeln('Заказчик: ', Zakazchik);
          writeln('Космодром: ', Kosmodrom);
          writeln('Результат: ', Rezultat);
          writeln;
          naideno := true;
        end;
      if not naideno then
        writeln('Запуски по заданному критерию не найдены.');
```

```

end;

procedure ZagruzkaIzFaila(var zapuskBase: ZapuskBase; var n: integer; filename: string);
var
  f: file of Zapusk;
  zapusk1: Zapusk;
  i: integer;
begin

```

```

AssignFile(f, filename);
Reset(f);
n := 0;
while not Eof(f) do
  begin
    Inc(n);
    Read(f, zapusk1);
  end;
SetLength(zapuskBase, n);
Reset(f);
for i := 0 to n - 1 do
  begin
    Read(f, zapuskBase[i]);
  end;
CloseFile(f);
end;

procedure SohranenieVFaile(zapuskBase: ZapuskBase; n: integer; filename: string);
var
  f: file of Zapusk;
  i: integer;
begin
  AssignFile(f, filename);
  Rewrite(f);
  for i := 0 to n - 1 do
    Write(f, zapuskBase[i]);
  CloseFile(f);
end;

var
  zapuskBaseObj: ZapuskBase;
  n: integer;
  filename: string = 'zapusk_data.dat';
  kriteriy: string;
  znachenie: string;
begin

  ZagruzkaIzFaile(zapuskBaseObj, n, filename);

  if n > 0 then
    begin
      writeln('Загруженные данные о запусках:');
      VypisZapuskov(zapuskBaseObj, n);
    end
  else
    writeln('Файл пуст или не найден.');
```

VvodZapuskov(zapuskBaseObj, n);

```
SohranenieVFaile(zapuskBaseObj, n, filename);

writeln('Введенные данные о запусках:');
VypisZapuskov(zapuskBaseObj, n);

writeln('Введите критерий поиска (Raketanositel, PoleznayaNagruzka, Zakazchik, Kosmodrom,
Rezultat):');

readln(kriteriy);

writeln('Введите значение для поиска:');
readln(znachenie);

PoiskZapuskovPoKriteriyam(zapuskBaseObj, n, kriteriy, znachenie);

read(n);

end.
```

## Тестирование реализации

Для данной реализации были использованы методы тестирования черным ящиком – метод эквивалентного разбиения – и белым ящиком – метод комбинаторного покрытия условий и решений. Каждая подпрограмма была протестирована отдельно.

- Подпрограмма «VvodZapusk»

Были выделены следующие эквивалентные классы возможных решений:

1. Сообщение было выведено корректно. Рассмотрены случаи для всех подусловий.

Результаты тестов представлены в таблице 1.

Таблица 1 – Тест реализации подпрограммы «VvodZapusk»

№	Входные данные	Выходные данные
1	10 апреля Буран 200 кг Роскосмос Байконур Отличный	Введенные данные о запусках: Дата запуска: 10 апреля Ракета-носитель: Буран Полезная нагрузка: 200 кг Заказчик: Роскосмос Космодром: Байконур Результат: Отличный

- Подпрограмма «VypisZapuskov»

Были выделены следующие эквивалентные классы возможных решений:

1. Сообщение было выведено корректно. Рассмотрены случаи для всех подусловий.

Результаты тестов представлены в таблице 2.

№	Входные данные	Выходные данные
1	2 10 апреля Левиофан 300 кг Роскосмос Байконур Классный 5 декабря Буран 200 кг NASA USA Отличный	Введенные данные о запусках:  Дата запуска: 10 апреля Ракета-носитель: Левиофан Полезная нагрузка: 300 кг Заказчик: Роскосмос Космодром: Байконур Результат: Классный  Дата запуска: 5 декабря Ракета-носитель: Буран Полезная нагрузка: 200 кг Заказчик: NASA Космодром: USA



		Результат: Отличный
--	--	---------------------

- Подпрограмма «PoiskZapuskovPoKriteriyam»

Были выделены следующие эквивалентные классы возможных решений:

2. Сообщение было выведено корректно. Рассмотрены случаи для всех подусловий.

Результаты тестов представлены в таблице 2.

№	Входные данные	Выходные данные
1	2 10 апреля Левиофан 300 кг Роскосмос Байконур Классный 5 декабря Буран 200 кг NASA USA Отличный PoleznayaNagruzka 200 кг	Дата запуска: 5 декабря Ракета-носитель: Буран Полезная нагрузка: 200 кг Заказчик: NASA Космодром: USA Результат: Отличный
2	2 10 апреля Левиофан 300 кг Роскосмос Байконур Классный 5 декабря Буран 200 кг NASA USA Отличный PoleznayaNagruzka 100 кг	Запуски по данному критерию не найдены

- Алгоритм всей программы

Программа содержит в себе подпрограммы «Ввод массива» и «Лучший кандидат», тестирование которых представлено в таблицах 1 и 2, поэтому приводить тестирование этой части программы ещё раз не имеет смысла. Таким образом, были выделены следующие эквивалентные классы возможных решений:

1. Сообщение было выведено корректно. Рассмотрены случаи для всех подусловий.
  2. Нарушение ввода – ввод дробных чисел или не чисел для уровня личных качеств или количества вакансий и кандидатов;
  3. Нарушение диапазона значений вводимого числа для вышеуказанных параметров;
- Результаты тестов представлены в таблице 3. Сообщения о системной ошибке представлены ввиду невозможности предусмотрения ошибок дополнительной проверкой согласно учебному плану.

№	Входные данные	Выходные данные
1	2 10 апреля Левиофан 300 кг Роскосмос Байконур Классный 5 декабря Буран 200 кг NASA USA Отличный PoleznayaNagruzka 100 кг	Введенные данные о запусках:  Дата запуска: 10 апреля Ракета-носитель: Левиофан Полезная нагрузка: 300 кг Заказчик: Роскосмос Космодром: Байконур Результат: Классный  Дата запуска: 5 декабря Ракета-носитель: Буран Полезная нагрузка: 200 кг Заказчик: NASA Космодром: USA Результат: Отличный  Запуски по данному критерию не найдены
	2 10 апреля Левиофан 300 кг Роскосмос Байконур Классный 5 декабря	Введенные данные о запусках:  Дата запуска: 10 апреля Ракета-носитель: Левиофан Полезная нагрузка: 300 кг Заказчик: Роскосмос Космодром: Байконур Результат: Классный

	Буран 200 кг NASA USA Отличный PoleznayaNagruzka 200 кг	Дата запуска: 5 декабря Ракета-носитель: Буран Полезная нагрузка: 200 кг Заказчик: NASA Космодром: USA Результат: Отличный  Дата запуска: 5 декабря Ракета-носитель: Буран Полезная нагрузка: 200 кг Заказчик: NASA Космодром: USA Результат: Отличный
2	2.5 10 апреля Левиофан 300 кг Роскосмос Байконур Классный 5 декабря Буран 200 кг NASA USA Отличный PoleznayaNagruzka 200 кг	Системная ошибка
3	true 10 апреля Левиофан 300 кг Роскосмос Байконур Классный 5 декабря Буран 200 кг NASA USA Отличный PoleznayaNagruzka 200 кг	Системная ошибка

## **ЗАКЛЮЧЕНИЕ**

В настоящей работе была составлена программа для создания базы данных запуска ракет, в которую вначале вводится количество запусков, потом по каждому полёту дату запуска, тип ракеты-носителя, полезную нагрузку, заказчика, космодром, результат запуска, после чего программа даёт пользователю выбор параметра, по которому можно отфильтровать запуск, после чего пользователь вводит этот параметр и ему выдаётся информация об определённом полёте.

Задачи:

- Составлена IDEF0-диаграмма проекта;
- Составлены блок-схемы алгоритмов;
- Алгоритмы реализованы на языке Object Pascal;
- Реализации алгоритмов протестированы, были подобраны классы ошибок.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Роскосмос:<https://www.roscosmos.ru/33/> (дата обращения 17.11.2024). Ракеты — носители.
2. Роскосмос:<https://www.roscosmos.ru/30/>(дата обращения 17.11.2024). Космодромы.
3. Роскосмос:<https://www.roscosmos.ru/30009>(дата обращения 17.11.2024). Статистические данные.