Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ

Иерархия наследования

Описать четыре класса которые последовательно наследуют друг друга, последовательными номерами классов 1,2,3,4.

Реализовать программу, в которой использовать единственный указатель на объект базового класса (номер класса 1).

Наследственность реализовать так, что можно было вызвать методы, принадлежащие объекту конкретного класса, только через объект данного класса.

В закрытом разделе каждого класса определены два свойства: строкового типа для наименования объекта и целого типа для значения определенного целочисленного выражения.

Описание каждого класса содержит один параметризированный конструктор с строковым и целочисленным параметром.

В реализации каждого конструктора объекта определяются значения закрытых свойств:

- Наименование объекта по шаблону: «значение строкового параметра»_«номер класса»;
- Целочисленного свойства значением выражения возведения в степень номера класса целочисленного значения параметра конструктора.

Еще в описании каждого класса определен метод с одинаковым наименованием для всех классов, реализующий вывод значений закрытых свойств класса.

В основной функции реализовать алгоритм:

1. Вводится идентификатор и натуральное число от 2 до 10.
2. Создать объект класса 4, используя параметризированный конструктор,

которому в качестве аргументов передаются введенный идентификатор и натуральное число.

3. Построчно, для всех объектов согласно наследственности, от объекта базового (класс 1) до производного объекта (класса 4) вывести наименование объекта класса и значение целочисленного свойства.

1.1 Описание входных данных

Первая строка:

`«идентификатор» «натуральное число»`

Пример ввода:

```
Object 2
```

1.2 Описание выходных данных

Построчно (четыре строки):

`«идентификатор»_«номер класса» «значение целочисленного свойства»`

Разделитель - 1 пробел.

Пример вывода:

```
Object_1 2
Object_2 4
Object_3 8
Object_4 16
```

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- Объект стандартного потока ввода с клавиатуры cin;
- Объект стандартного потока вывода на экран cout;
- Условный оператор if..else.

Класс Obj_1:

- свойства/поля:
    - поле Наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
    - поле Значение выражения:
        - наименование — number;
        - тип — int;
        - модификатор доступа — private;
- функционал:
    - метод Obj_1 — Конструктор параметризованный;
    - метод output — Метод вывода значений свойств.

Класс Obj_2:

- свойства/поля:
    - поле Наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
    - поле Значение выражения:
        - наименование — number;

        - тип — int;
        - модификатор доступа — private;
- функционал:
    - метод Obj_2 — Конструктор параметризованный;
    - метод output — Метод вывода значений свойств.

Класс Obj_3:

- свойства/поля:
    - поле Наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
    - поле Значение выражения:
        - наименование — number;
        - тип — int;
        - модификатор доступа — private;
- функционал:
    - метод Obj_3 — Конструктор параметризованный;
    - метод output — Метод вывода значений свойств.

Класс Obj_4:

- свойства/поля:
    - поле Наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
    - поле Значение выражения:
        - наименование — number;
        - тип — int;

        - модификатор доступа — private;
- функционал:
    - метод Obj_4 — Конструктор параметризованный;
    - метод output — Метод вывода значений свойств.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	Obj_1			Базовый класс	
		Obj_2	private		2
2	Obj_2			Производный класс	
		Obj_3	private		3
3	Obj_3			Производный класс	
		Obj_4	private		4
4	Obj_4			Производный класс	

3 ОПИСАНИЕ АЛГОРИТМОВ

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

3.1 Алгоритм конструктора класса Obj_1

Функционал: Конструктор параметризованный.

Параметры: Строка name и целочисленная переменная number для инициализации свойств.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса Obj_1

№	Предикат	Действия	№ перехода
1		this->name инициализируется значением строки name + "_1"	2
2		this->number инициализируется значением number в степени 1	∅

3.2 Алгоритм метода output класса Obj_1

Функционал: Метод вывода значений свойств.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода output класса Obj_1

№	Предикат	Действия	№ перехода
1		Вывод на экран "(this->name) пробел (this->number)"	∅

3.3 Алгоритм конструктора класса Obj_2

Функционал: Конструктор параметризованный.

Параметры: Строка name и целочисленная переменная number для инициализации свойств.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса Obj_2

№	Предикат	Действия	№ перехода
1		this->name инициализируется значением строки name + "_2"	2
2		this->number инициализируется значением number в степени 2	∅

3.4 Алгоритм метода output класса Obj_2

Функционал: Метод вывода значений свойств.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода output класса Obj_2

№	Предикат	Действия	№ перехода
1		Вывод на экран "(this->name) пробел (this->number)"	∅

3.5 Алгоритм конструктора класса Obj_3

Функционал: Конструктор параметризованный.

Параметры: Строка name и целочисленная переменная number для инициализации свойств.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса Obj_3

№	Предикат	Действия	№ перехода
1		this->name инициализируется значением строки name + "_3"	2
2		this->number инициализируется значением number в степени 3	∅

3.6 Алгоритм метода output класса Obj_3

Функционал: Метод вывода значений свойств.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода output класса Obj_3

№	Предикат	Действия	№ перехода
1		Вывод на экран "(this->name) пробел (this->number)"	∅

3.7 Алгоритм конструктора класса Obj_4

Функционал: Конструктор параметризованный.

Параметры: Строка name и целочисленная переменная number для инициализации свойств.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса Obj_4

№	Предикат	Действия	№ перехода
1		this->name инициализируется значением строки name + "_4"	2
2		this->number инициализируется значением number в степени 4	∅

3.8 Алгоритм метода output класса Obj_4

Функционал: Метод вывода значений свойств.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода output класса Obj_4

№	Предикат	Действия	№ перехода
1		Вывод на экран "(this->name) пробел (this->number)"	∅

3.9 Алгоритм функции main

Функционал: Работа с иерархией наследования.

Параметры: нет.

Возвращаемое значение: Целочисленное значение.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление строковой перемеенной name и целочисленой переменной number	2
2		Ввод с клавиатуры значения name и number	3
3	number>=2 и number<=10	Создание указателя obj на объект класса Obj_4 с параметрами name и number	4
			∅
4		Вызов метода output родительского класса Obj_1 для объекта с адресом obj	5
5		Вызов метода output родительского класса Obj_2 для объекта с адресом obj	6

№	Предикат	Действия	№ перехода
6		Вызов метода output родительского класса Obj_3 для объекта с адресом obj	7
7		Вызов метода output класса Obj_4 для объекта с адресом obj	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-2.

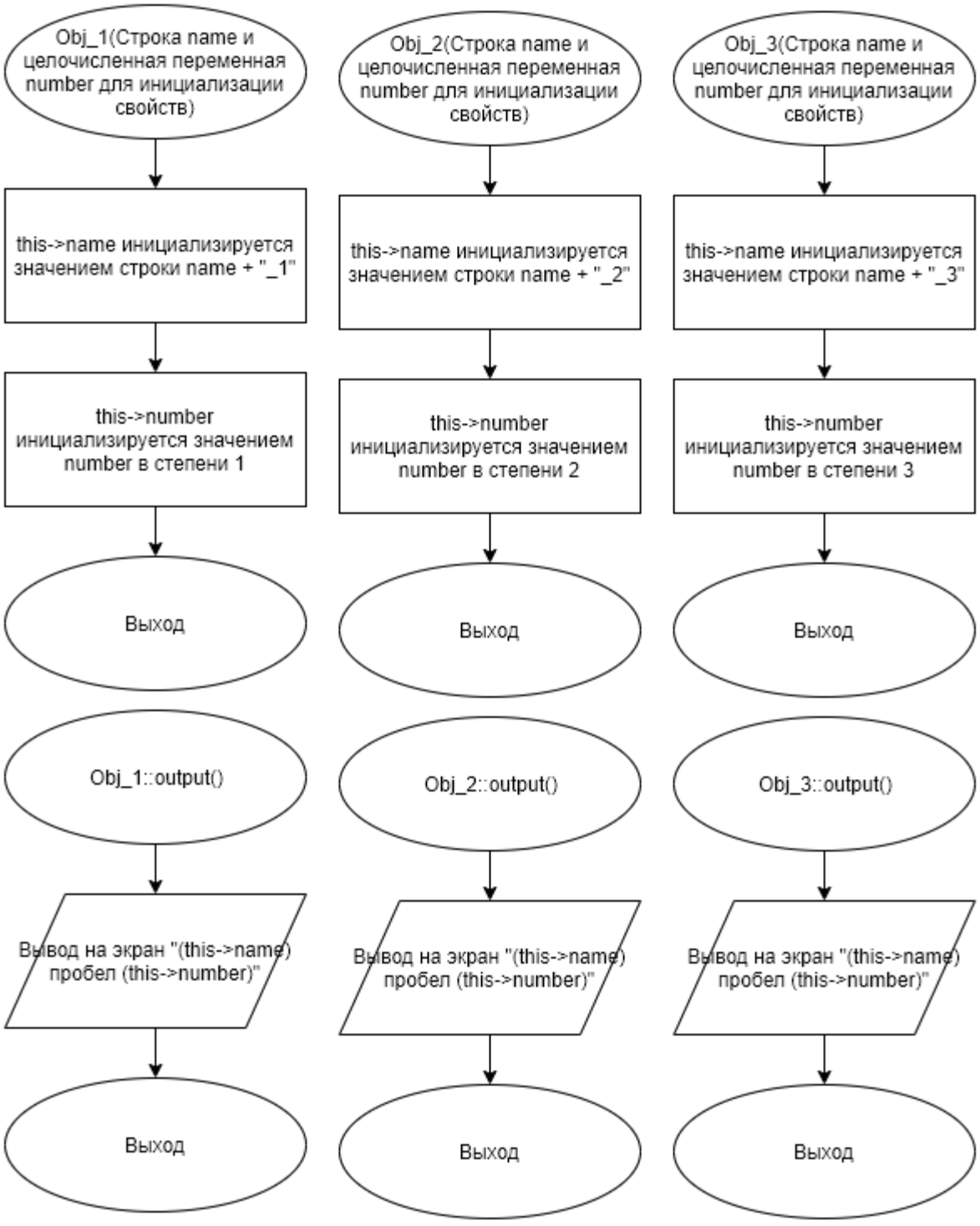

Рисунок 1 – Блок-схема алгоритма

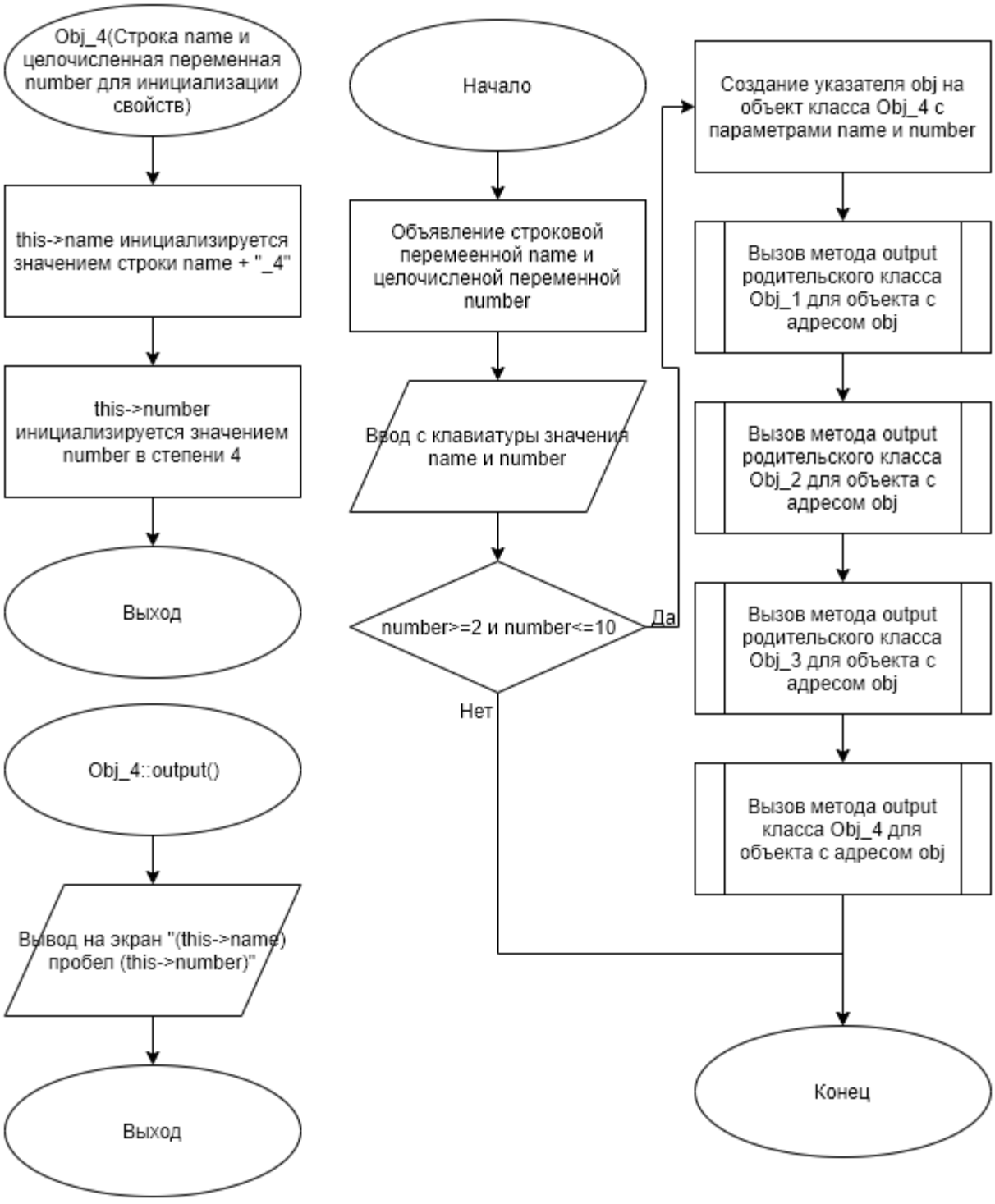

Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "Obj_4.h"

using namespace std;

int main()
{
   string name;
   int number;
   cin >> name >> number;
   if (number >= 2 && number <= 10) {
      Obj_4* obj = new Obj_4(name, number);
      ((Obj_1*)obj)->output();
      ((Obj_2*)obj)->output();
      ((Obj_3*)obj)->output();
      obj->output();
   }
   else {
      exit(0);
   }
}
```

5.2 Файл Obj_1.cpp

Листинг 2 – Obj_1.cpp

```
#include "Obj_1.h"

Obj_1::Obj_1(string name, int number)
{
   this->name = name + "_1";
   this->number = pow(number, 1);
```

```
}

void Obj_1::output()
{
   cout << this->name << " " << this->number << endl;
}
```

5.3 Файл Obj_1.h

Листинг 3 – Obj_1.h

```
#ifndef __OBJ_1__H
#define __OBJ_1__H
#include <iostream>
#include <string>
#include <cmath>

using namespace std;

class Obj_1
{
private:
   string name;
   int number;

public:
   Obj_1(string name, int number);
   void output();
};

#endif
```

5.4 Файл Obj_2.cpp

Листинг 4 – Obj_2.cpp

```
#include "Obj_2.h"

Obj_2::Obj_2(string name, int number): Obj_1(name, number)
{
   this->name = name + "_2";
   this->number = pow(number, 2);
}

void Obj_2::output()
```

```
{
   cout << this->name << " " << this->number << endl;
}
```

5.5 Файл Obj_2.h

Листинг 5 – Obj_2.h

```
#ifndef __OBJ_2__H
#define __OBJ_2__H
#include <iostream>
#include <string>
#include <cmath>
#include "Obj_1.h"

using namespace std;

class Obj_2: private Obj_1
{
private:
   string name;
   int number;

public:
   Obj_2(string name, int number);
   void output();
};


#endif
```

5.6 Файл Obj_3.cpp

Листинг 6 – Obj_3.cpp

```
#include "Obj_3.h"

Obj_3::Obj_3(string name, int number): Obj_2(name, number)
{
   this->name = name + "_3";
   this->number = pow(number, 3);
}

void Obj_3::output()
{
```

```
   cout << this->name << " " << this->number << endl;
}
```

5.7 Файл Obj_3.h

Листинг 7 – Obj_3.h

```
#ifndef __OBJ_3__H
#define __OBJ_3__H
#include <iostream>
#include <string>
#include <cmath>
#include "Obj_2.h"

using namespace std;

class Obj_3: private Obj_2
{
private:
   string name;
   int number;

public:
   Obj_3(string name, int number);
   void output();
};



#endif
```

5.8 Файл Obj_4.cpp

Листинг 8 – Obj_4.cpp

```
#include "Obj_4.h"

Obj_4::Obj_4(string name, int number): Obj_3(name, number)
{
   this->name = name + "_4";
   this->number = pow(number, 4);
}

void Obj_4::output()
{
```

```
   cout << this->name << " " << this->number << endl;
}
```

5.9 Файл Obj_4.h

Листинг 9 – Obj_4.h

```
#ifndef __OBJ_4__H
#define __OBJ_4__H
#include <iostream>
#include <string>
#include <cmath>
#include "Obj_3.h"

using namespace std;

class Obj_4: private Obj_3
{
private:
   string name;
   int number;

public:
   Obj_4(string name, int number);
   void output();
};



#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 11.

Таблица 11 – Результат тестирования программы

Входные данные	**Ожидаемые выходные данные**	**Фактические выходные данные**

`Object 2`	`Object_1 2`
	`Object_2 4`
	`Object_3 8`
	`Object_4 16`
		`Object_1 2`
		`Object_2 4`
		`Object_3 8`
		`Object_4 16`

`qq 3`	`qq_1 3`
	`qq_2 9`
	`qq_3 27`
	`qq_4 81`
		`qq_1 3`
		`qq_2 9`
		`qq_3 27`
		`qq_4 81`

`wewe 1`		
`eror 11`		

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19 Единая система программной документации.

2. Методическое пособие студента для выполнения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrora.ru/student/files/methodichescoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).

3. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).

4. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2019. — 624 с.

5. Видео лекции по курсу «Объектно-ориентированное программирование» [Электронный ресурс]. АСО «Аврора».

6. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие /Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).