Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В закрытом доступе имеется массив целого типа и поле его длины. Количество элементов массива четное и больше двух. Объект имеет функциональность:

- Конструктор по умолчанию, вначале работы выдает сообщение;
- Параметризированный конструктор, передается целочисленный параметр. Параметр должен иметь значение больше 2 и быть четным. Вначале работы выдает сообщение;
- Конструктор копии, обеспечивает создание копии объекта в новой области памяти. Вначале работы выдает сообщение;
- Метод деструктор, который в начале работы выдает сообщение;
- Метод который создает целочисленный массив в закрытой области, согласно ранее заданной размерности.
- Метод ввода данных для созданного массива;
- Метод 1, который суммирует значения очередной пары элементов и сумму присваивает первому элементу пары. Например, пусть массив состоит из элементов {1,2,3,4}. В результате суммирования пар получим массив {3,2,7,4};
- Метод 2, который умножает значения очередной пары элементов и результат присваивает первому элементу пары. Например, пусть массив состоит из элементов {1,2,3,4}. В результате умножения пар получим массив {2,2,12,4};
- Метод который, суммирует значения элементов массива и возвращает это значение;
- Метод последовательного вывода содержимого элементов массива,

которые разделены тремя пробелами.

Разработать функцию func, которая имеет один целочисленный параметр, содержащий размерность массива. В функции должен быть реализован алгоритм:

1. Создание локального объекта с использованием параметризированного конструктора.
2. Возврат созданного локального объекта.

В основной функции реализовать алгоритм:

1. Ввод размерности массива.
2. Если размерность массива некорректная, вывод сообщения и завершить работу алгоритма.
3. Вывод значения размерности массива.
4. Создание первого объекта.
5. Присвоение первому объекту результата работы функции func с аргументом, содержащим значение размерности массива.
6. Для первого объекта вызов метода создания массива.
7. Для первого объекта вызов метода ввода данных массива.
8. Для первого объекта вызов метода 2.
9. Инициализация второго объекта первым объектом.
10. Вызов метода 1 для второго объекта.
11. Вывод содержимого массива первого объекта.
12. Вывод суммы элементов массива первого объекта.
13. Вывод содержимого массива второго объекта.
14. Вывод суммы элементов массива второго объекта.

1.1 Описание входных данных

Первая строка:

```
«Целое число»
```

Вторая строка:

```
«Целое число» «Целое число» . . .
```

Пример:

```
4
3 5 1 2
```

1.2 Описание выходных данных

Если введенная размерность массива допустима, то в первой строке выводится это значение:

```
«Целое число»
```

Если введенная размерность массива не больше двух или нечетная, то в первой строке выводится некорректное значение и вопросительный знак:

```
«Целое число»?
```

Конструктор по умолчанию в начале работы с новой строки выдает сообщение:

```
Default constructor
```

Параметризированный конструктор в начале работы с новой строки выдает сообщение:

```
Constructor set
```

Конструктор копии в начале работы с новой строки выдает сообщение:

```
Copy constructor
```

Деструктор в начале работы с новой строки выдает сообщение:

```
Destructor
```

Метод последовательного вывода содержимого элементов массива, с новой строки выдает:

```
«Целое число»   «Целое число»   «Целое число»   . . .
```

Пример вывода:

```
4
Default constructor
Constructor set
Destructor
Copy constructor
15   5   2   2
24
20   5   4   2
31
Destructor
Destructor
```

2 МЕТОД РЕШЕНИЯ

Класс Item:

- функционал:
    - метод CreateA — Создание целочисленного массива;
    - метод OutputA — Последовательный вывод содержимого элементов массива.

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 метода OutputA класса Item

Функционал: Последовательный вывод содержимого элементов массива, разделенные тремя пробелами.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода OutputA класса Item

№	Предикат	Действия	№ перехода
1	Целочисленная переменная i от 1 до s, шаг 1	Вывод элемента массива a с индексом i	2
			∅
2	Элемент массива последний		3
		Вывод три пробела	3
3		Икремент i	1

3.2 Алгоритм метода CreateA класса Item

Функционал: Создание целочисленного массива.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода CreateA класса Item

№	Предикат	Действия	№ перехода
1		Созание целочисленного массива заданного размера	∅

3.3 Алгоритм конструктора класса Item

Функционал: Создание объекта на основе класса Item.

Параметры: int s.

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса Item

№	Предикат	Действия	№ перехода
1	Размер массива нечетный или меньше 2	Вывод значения переменной s и "?"	∅
		Создание массива a размера s	2
2		Вывод "Constructor set"	3
3		Полю s текщего класса присвоить значение параметра s	∅

3.4 Алгоритм функции F

Функционал: Создание локального объекта.

Параметры: int size.

Возвращаемое значение: Возврат значения созданного объекта класса Item.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции F

№	Предикат	Действия	№ перехода
1		Создание объекта с использованием параметризированного конструктора	2

№	Предикат	Действия	№ перехода
2		Возврат созданного объекта	∅

3.5 Алгоритм функции main

Функционал: Основной алгоритм работы программы.

Параметры: нет.

Возвращаемое значение: Целое, индикатор корректности завершения программы.

Алгоритм функции представлен в таблице 5.

Таблица 5 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной s	2
2		Ввод значения переменной s	3
3	Значение переменной s меньше 2 или нечётное	Вывод значения переменной s и "?"	∅
		Вывод значения переменной s и символ переходна на новую строку	4
4		Создание объекта o1 класса Item	5
5		Объекту o1 присвоить значение результат работы функции F с аргументов в виде значения переменной s	6
6		Вызов метода CreateA объекта o класса Item	7
7		Вызов метода InputA объекта o класса Item	8
8		Вызов метода Multi у объекта o1 класса Item	9
9		Инициализация объекта o2 класса Item объектом o1 класса Item	10
10		Вызов метода twoPlus у объекта o2 класса Item	11
11		Вызов метода OutputA у объекта o1 класса Item	12

№	Предикат	Действия	№ перехода
12		Вызов метода Plus у объекта o1 класса Item	13
13		Вызов метода OutputA у объекта o2 класса Item	14
14		Вызов метода Plus у объекта o2 класса Item	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-3.

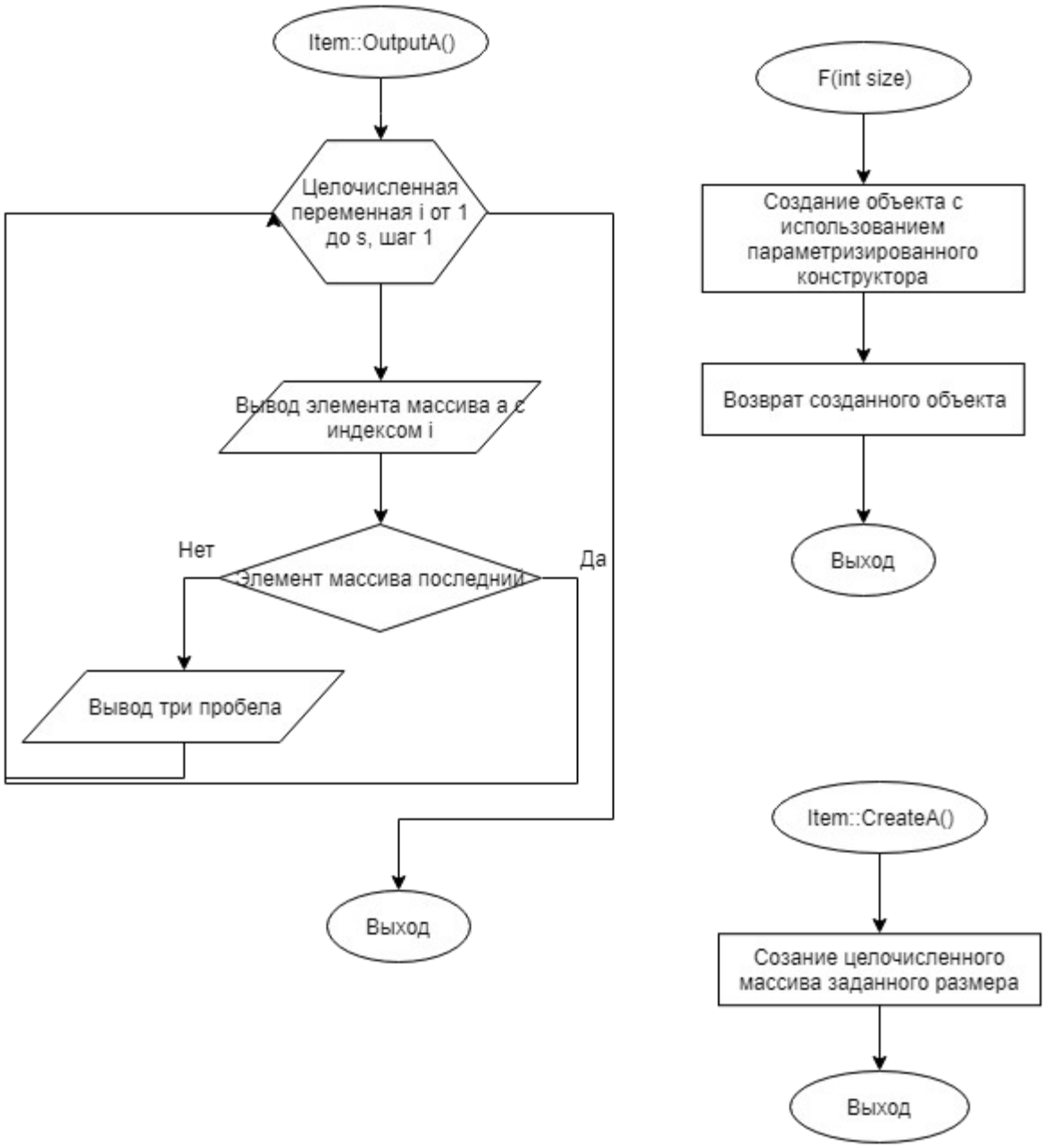

Рисунок 1 – Блок-схема алгоритма

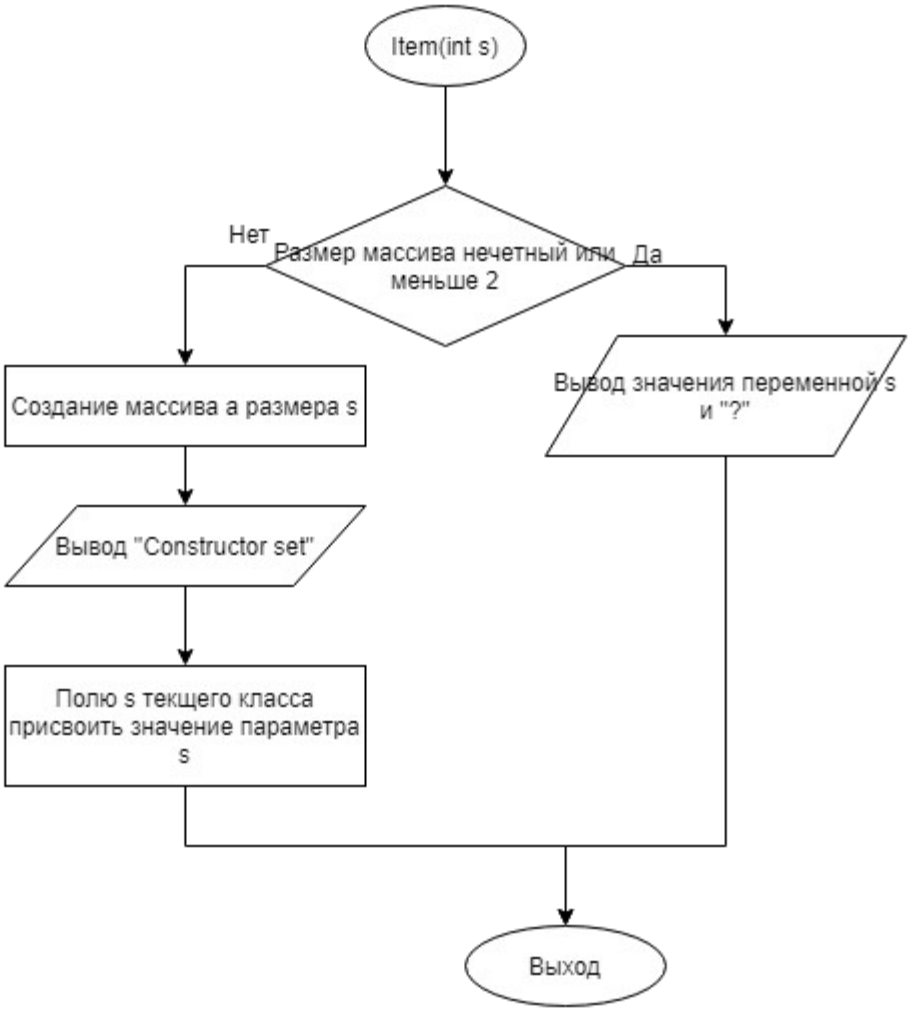

Рисунок 2 – Блок-схема алгоритма

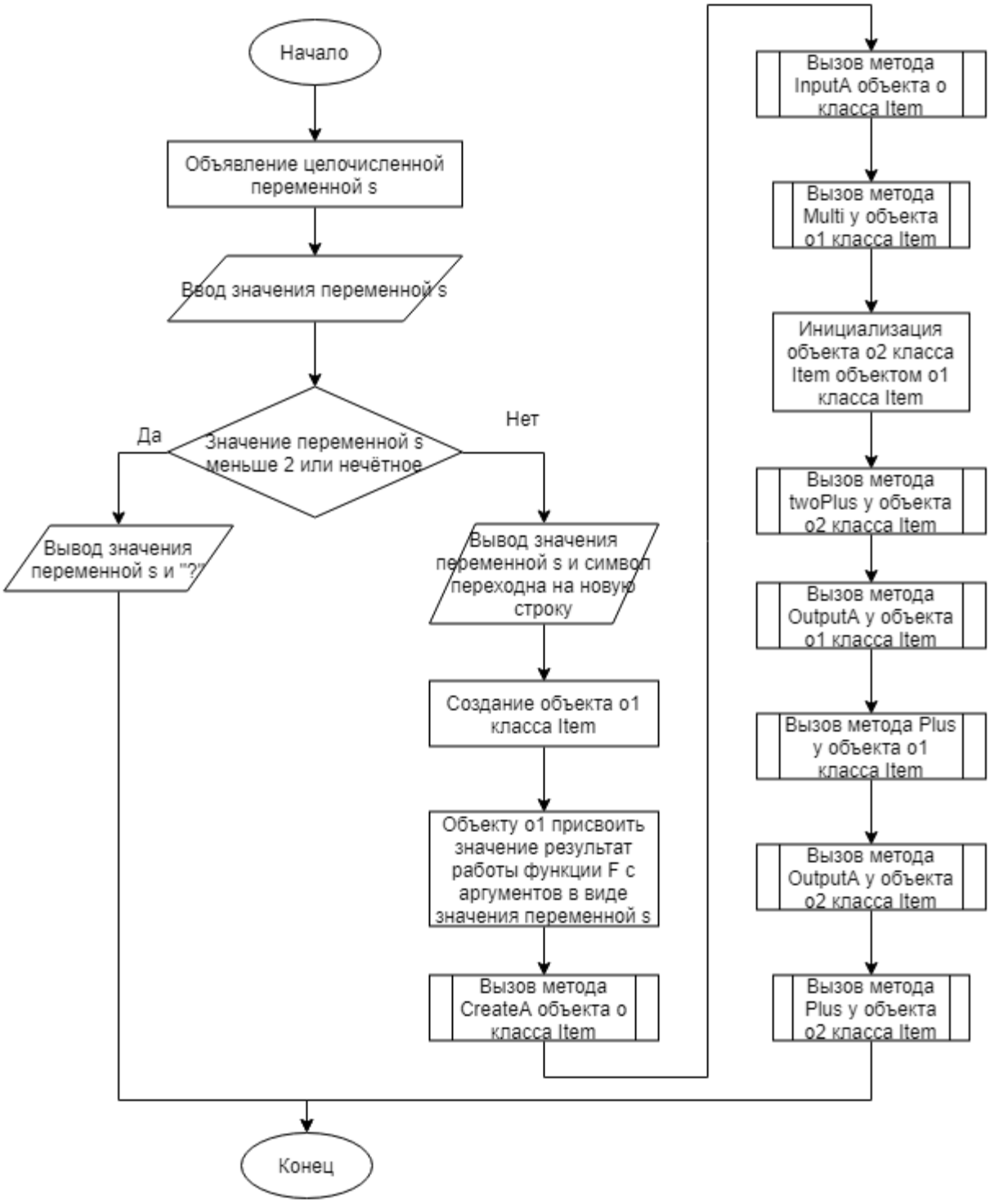

Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Item.cpp

Листинг 1 – Item.cpp

```
#include "Item.h"
#include <iostream>
#include <string>

using namespace std;

Item::Item()
{
   cout << "Default constructor" << endl;
}

Item::Item(int s)
{
   if (s <= 2 || s % 2 != 0)
   {
      cout << s << "?";
   }

   a = new int[s];
   cout << "Constructor set" << endl;
   this -> s = s;
}

Item::Item(const Item & o)
{
   cout << "Copy constructor" << endl;
   s = o.s;
   a = new int[s];

   for (int i = 0; i < s; i++)
      a[i] = o.a[i];
}

void Item::CreateA()
{
   a = new int[s];
}

void Item::Completion()
```

```
{
   for (int i = 0; i < s; i++)
   {
      cin >> a[i];
   }
}

Item::~Item()
{
   static int count = 0;
   cout << "Destructor";

   if (a != nullptr)
      delete[]a;
   count++;

   if (count == 1 || count == 2)
   {
      cout << endl;
   }
}

void Item::Multi()
{
   for (int i = 0; i < s; i = i+2)
   {
      a[i] = a[i] * a[i+1];
   }
}

void Item::twoPlus()
{
   for (int i = 0; i < s; i = i+2)
   {
      a[i] = a[i] + a[i+1];
   }
}

int Item::Plus()
{
   int sum = 0;
   for (int i = 0; i < s; i++)
   {
      sum = sum + a[i];
   }
   cout << sum << endl;
   return sum;
}

void Item::OutputA()
{
   for (int i = 0; i < s; i++)
   {
      cout << a[i];
      if (i < s - 1)
```

```
            {
               cout << "   ";
            }
         }
         cout << endl;
      }
```

5.2 Файл Item.h

Листинг 2 – Item.h

```
      #ifndef __ITEM__H
      #define __ITEM__H
      #include <iostream>

      using namespace std;

      class Item
      {
         private:
            int* a;
            int s;
         public:
            ~Item();
            Item();
            Item(int s);
            Item(const Item& o);
            void Completion();
            void OutputA();
            void Multi();
            void twoPlus();
            int Plus();
            void CreateA();
      };
      #endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
      #include "Item.h"
      #include <iostream>
      using namespace std;
      Item F(int s)
      {
```

```
        Item o(s);
        return o;
    }
    int main()
    {
        int s;
        cin >> s;
        if ((s <= 2) || ( s % 2 != 0))
        {
            cout << s << "?";
            return 0;
        }
        cout << s << endl;
        Item o1;
        o1 = F(s);

        o1.CreateA();
        o1.Completion();
        o1.Multi();

        Item o2 = o1;
        o2.twoPlus();

        o1.OutputA();

        o1.Plus();
        o2.OutputA();

        o2.Plus();
        return 0;
    }
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 6.

Таблица 6 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
4 3 5 1 2	4 Default constructor Constructor set Destructor Copy constructor 15 5 2 2 24 20 5 4 2 31 Destructor Destructor	4 Default constructor Constructor set Destructor Copy constructor 15 5 2 2 24 20 5 4 2 31 Destructor Destructor

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).