

Контрольная работа №1

Тема «Основы построения диаграмм UML»

Вариант №1

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y;  
double z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextDouble();  
if (x!=15 && y>=65)  
    z=x+y;  
else if (x<=150)  
    z=y-x;  
else {  
    z++;  
    x-=y;  
}  
System.out.println("x="+x);  
System.out.println("z="+z);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение варианта использования *B1*.

Актер *A2* инициирует выполнение вариантов использования *B2* и *B3*.

Варианты использования *B1* и *B2* являются потомками варианта использования *B4*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B3*.

Вариант №2

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y, z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextInt();  
while (x<800) {  
    if (y<=13 || z>75)  
        y+=x;  
    x+=25;  
}  
System.out.println("x="+x);  
System.out.println("y="+y);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение вариантов использования *B3* и *B4*.

Актер *A1* является потомком актера *A2*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B2*.

Вариант №3

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y;  
double z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextDouble();  
do {  
    if (y>=7 && z<=14)  
        z+=2;  
    x-=3;  
} while (x>70);  
System.out.println("x="+x);  
System.out.println("z="+z);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение варианта использования *B3*.

Актер *A1* является родителем актера *A2*.

Вариант использования *B4* является включаемым по отношению к варианту использования *B3*.

Вариант №4

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int n, s=0;
Scanner sc=new Scanner(System.in);
Random r=new Random();
System.out.print("n=");
n=sc.nextInt();
int[] mas=new int[n];
for (int i=0; i<n; i++) {
    mas[i]=r.nextInt(100);
    System.out.print(mas[i]+" ");
    if (mas[i]%5!=0 || mas[i]<18)
        s+=mas[i];
}
System.out.println("\ns="+s);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение варианта использования *B1*.

Актер *A2* инициирует выполнение вариантов использования *B2* и *B3*.

Варианты использования *B1* и *B2* являются потомками варианта использования *B4*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B3*.

Вариант №5

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int n, s=0;
Scanner sc=new Scanner(System.in);
Random r=new Random();
System.out.print("n=");
n=sc.nextInt();
int[] mas=new int[n];
for (int i=0; i<n; i++) {
    mas[i]=r.nextInt(100);
    System.out.print(mas[i]+" ");
    if (mas[i]%4==0 && mas[i]>43)
        s+=mas[i];
}
System.out.println("\ns="+s);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение вариантов использования *B3* и *B4*.

Актер *A1* является потомком актера *A2*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B2*.

Вариант №6

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y;  
double z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextDouble();  
if (x!=30 && y>=50)  
    z=x+y;  
else if (x<199)  
    z=y-x;  
else {  
    z++;  
    x-=y;  
}  
System.out.println("x="+x);  
System.out.println("z="+z);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение варианта использования *B3*.

Актер *A1* является родителем актера *A2*.

Вариант использования *B4* является включаемым по отношению к варианту использования *B3*.

Вариант №7

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y, z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextInt();  
while (x<450) {  
    if (y>27 || z>=49)  
        y+=x;  
    x+=21;  
}  
System.out.println("x="+x);  
System.out.println("y="+y);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение варианта использования *B1*.

Актер *A2* инициирует выполнение вариантов использования *B2* и *B3*.

Варианты использования *B1* и *B2* являются потомками варианта использования *B4*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B3*.

Вариант №8

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y;  
double z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextDouble();  
do {  
    if (y>0 && z<350)  
        z++;  
    x+=7;  
} while (x<=62);  
System.out.println("x="+x);  
System.out.println("z="+z);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение вариантов использования *B3* и *B4*.

Актер *A1* является потомком актера *A2*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B2*.

Вариант №9

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int n, s=0;
Scanner sc=new Scanner(System.in);
Random r=new Random();
System.out.print("n=");
n=sc.nextInt();
int[] mas=new int[n];
for (int i=0; i<n; i++) {
    mas[i]=r.nextInt(100);
    System.out.print(mas[i]+" ");
    if (mas[i]%2==1 || mas[i]>=31)
        s+=mas[i];
}
System.out.println("\ns="+s);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение варианта использования *B3*.

Актер *A1* является родителем актера *A2*.

Вариант использования *B4* является включаемым по отношению к варианту использования *B3*.

Вариант №10

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int n, s=0;
Scanner sc=new Scanner(System.in);
Random r=new Random();
System.out.print("n=");
n=sc.nextInt();
int[] mas=new int[n];
for (int i=0; i<n; i++) {
    mas[i]=r.nextInt(100);
    System.out.print(mas[i]+" ");
    if (mas[i]%3==2 && mas[i]<78)
        s+=mas[i];
}
System.out.println("\ns="+s);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение варианта использования *B1*.

Актер *A2* инициирует выполнение вариантов использования *B2* и *B3*.

Варианты использования *B1* и *B2* являются потомками варианта использования *B4*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B3*.

Вариант №11

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y;  
double z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextDouble();  
if (x!=20 && y>=70)  
    z=x+y;  
else if (x<150)  
    z=y-x;  
else {  
    z++;  
    x-=y;  
}  
System.out.println("x="+x);  
System.out.println("z="+z);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение вариантов использования *B3* и *B4*.

Актер *A1* является потомком актера *A2*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B2*.

Вариант №12

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y;  
double z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextDouble();  
if (x!=10 && y>60)  
    z=x+y;  
else if (x<=100)  
    z=y-x;  
else {  
    z++;  
    x-=y;  
}  
System.out.println("x="+x);  
System.out.println("z="+z);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение варианта использования *B3*.

Актер *A1* является родителем актера *A2*.

Вариант использования *B4* является включаемым по отношению к варианту использования *B3*.

Вариант №13

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y, z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextInt();  
while (x<=500) {  
    if (y<20 || z>90)  
        y+=x;  
    x+=15;  
}  
System.out.println("x="+x);  
System.out.println("y="+y);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение варианта использования *B1*.

Актер *A2* инициирует выполнение вариантов использования *B2* и *B3*.

Варианты использования *B1* и *B2* являются потомками варианта использования *B4*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B3*.

Вариант №14

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y;  
double z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextDouble();  
do {  
    if (y>=3 && z<12)  
        z++;  
    x-=2;  
} while (x>=10);  
System.out.println("x="+x);  
System.out.println("z="+z);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение вариантов использования *B3* и *B4*.

Актер *A1* является потомком актера *A2*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B2*.

Вариант №15

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int n, s=0;
Scanner sc=new Scanner(System.in);
Random r=new Random();
System.out.print("n=");
n=sc.nextInt();
int[] mas=new int[n];
for (int i=0; i<n; i++) {
    mas[i]=r.nextInt(100);
    System.out.print(mas[i]+" ");
    if (mas[i]%2!=0 || mas[i]<10)
        s+=mas[i];
}
System.out.println("\ns="+s);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение варианта использования *B1*.

Актер *A2* инициирует выполнение вариантов использования *B2* и *B3*.

Варианты использования *B1* и *B2* являются потомками варианта использования *B4*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B3*.

Вариант №16

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int n, s=0;
Scanner sc=new Scanner(System.in);
Random r=new Random();
System.out.print("n=");
n=sc.nextInt();
int[] mas=new int[n];
for (int i=0; i<n; i++) {
    mas[i]=r.nextInt(100);
    System.out.print(mas[i]+" ");
    if (mas[i]%2==0 && mas[i]>45)
        s+=mas[i];
}
System.out.println("\ns="+s);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение варианта использования *B3*.

Актер *A1* является родителем актера *A2*.

Вариант использования *B4* является включаемым по отношению к варианту использования *B3*.

Вариант №17

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y;  
double z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextDouble();  
if (x!=25 && y>=55)  
    z=x+y;  
else if (x<210)  
    z=y-x;  
else {  
    z++;  
    x-=y;  
}  
System.out.println("x="+x);  
System.out.println("z="+z);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение вариантов использования *B3* и *B4*.

Актер *A1* является потомком актера *A2*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B2*.

Вариант №18

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y, z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextInt();  
while (x<330) {  
    if (y>=19 || z>11)  
        y+=x;  
    x+=15;  
}  
System.out.println("x="+x);  
System.out.println("y="+y);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение варианта использования *B3*.

Актер *A1* является родителем актера *A2*.

Вариант использования *B4* является включаемым по отношению к варианту использования *B3*.

Вариант №19

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y;  
double z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextDouble();  
do {  
    if (y>44 && z<=3)  
        z++;  
    x+=6;  
} while (x<85);  
System.out.println("x="+x);  
System.out.println("z="+z);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение варианта использования *B1*.

Актер *A2* инициирует выполнение вариантов использования *B2* и *B3*.

Варианты использования *B1* и *B2* являются потомками варианта использования *B4*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B3*.

Вариант №20

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int n, s=0;
Scanner sc=new Scanner(System.in);
Random r=new Random();
System.out.print("n=");
n=sc.nextInt();
int[] mas=new int[n];
for (int i=0; i<n; i++) {
    mas[i]=r.nextInt(100);
    System.out.print(mas[i]+" ");
    if (mas[i]%2==1 || mas[i]>15)
        s+=mas[i];
}
System.out.println("\ns="+s);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение вариантов использования *B3* и *B4*.

Актер *A1* является потомком актера *A2*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B2*.

Вариант №21

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int n, s=0;
Scanner sc=new Scanner(System.in);
Random r=new Random();
System.out.print("n=");
n=sc.nextInt();
int[] mas=new int[n];
for (int i=0; i<n; i++) {
    mas[i]=r.nextInt(100);
    System.out.print(mas[i]+" ");
    if (mas[i]%3==0 && mas[i]<39)
        s+=mas[i];
}
System.out.println("\ns="+s);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение варианта использования *B3*.

Актер *A1* является родителем актера *A2*.

Вариант использования *B4* является включаемым по отношению к варианту использования *B3*.

Вариант №22

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y;
double z;
Scanner sc=new Scanner(System.in);
System.out.print("x=");
x=sc.nextInt();
System.out.print("y=");
y=sc.nextInt();
System.out.print("z=");
z=sc.nextDouble();
if (x!=95 && y>105)
    z=x+y;
else if (x<150)
    z=y-x;
else {
    z++;
    x-=y;
}
System.out.println("x="+x);
System.out.println("z="+z);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение варианта использования *B1*.

Актер *A2* инициирует выполнение вариантов использования *B2* и *B3*.

Варианты использования *B1* и *B2* являются потомками варианта использования *B4*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B3*.

Вариант №23

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y, z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextInt();  
while (x<=950) {  
    if (y<=23 || z>85)  
        y+=x;  
    x+=25;  
}  
System.out.println("x="+x);  
System.out.println("y="+y);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение вариантов использования *B3* и *B4*.

Актер *A1* является потомком актера *A2*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B2*.

Вариант №24

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int x, y;  
double z;  
Scanner sc=new Scanner(System.in);  
System.out.print("x=");  
x=sc.nextInt();  
System.out.print("y=");  
y=sc.nextInt();  
System.out.print("z=");  
z=sc.nextDouble();  
do {  
    if (y>70 && z<=140)  
        z+=4;  
        x-=3;  
} while (x>70);  
System.out.println("x="+x);  
System.out.println("z="+z);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение варианта использования *B3*.

Актер *A1* является родителем актера *A2*.

Вариант использования *B4* является включаемым по отношению к варианту использования *B3*.

Вариант №25

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int n, s=0;
Scanner sc=new Scanner(System.in);
Random r=new Random();
System.out.print("n=");
n=sc.nextInt();
int[] mas=new int[n];
for (int i=0; i<n; i++) {
    mas[i]=r.nextInt(100);
    System.out.print(mas[i]+" ");
    if (mas[i]%5!=0 || mas[i]>=80)
        s+=mas[i];
}
System.out.println("\ns="+s);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение варианта использования *B1*.

Актер *A2* инициирует выполнение вариантов использования *B2* и *B3*.

Варианты использования *B1* и *B2* являются потомками варианта использования *B4*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B3*.

Вариант №26

Задача №1. Построить алгоритм работы следующей программы на языке Java в виде диаграммы деятельности UML, описывающей ввод требуемых данных, необходимые вычисления и вывод информации на экран:

```
int n, s=0;
Scanner sc=new Scanner(System.in);
Random r=new Random();
System.out.print("n=");
n=sc.nextInt();
int[] mas=new int[n];
for (int i=0; i<n; i++) {
    mas[i]=r.nextInt(100);
    System.out.print(mas[i]+" ");
    if (mas[i]%4==0 && mas[i]>=36)
        s+=mas[i];
}
System.out.println("\ns="+s);
```

Задача №2. Нарисовать диаграмму вариантов использования UML по данному описанию:

Актер *A1* инициирует выполнение вариантов использования *B1* и *B2*.

Актер *A2* инициирует выполнение вариантов использования *B3* и *B4*.

Актер *A1* является потомком актера *A2*.

Вариант использования *B5* является расширяющим по отношению к варианту использования *B2*.