


```

return s;
static void Main(string[] args)
{
    //Randomстандартты датчигін
    инициализациялау random =
    newRandom();
    for (int i = 0; i < R; i++)
    {
        // Бірінші кездейсоқ санның
        түрленуіRand[i] =
        random.NextDouble();
    }
    for (int i = 0; i < N MAX; i++)
    {
        double x = Rnd();
    }
}

```

2.2.1. Кездейсоқ санның қиыстыру генераторы

Циклді қайталанатын тізбекті модельдеу кезінде қолдануды болдырмас үшін (период ұзындығын ұзарту үшін) кездейсоқ санның қиыстыру генераторлары кеңінен қолданылады.

Екі қарапайым генераторды қолданатын осындай генератордың қарапайым мысалын қарастырайық. Олардың көмегімен кездейсоқ санның екі тізбегін алады: біреуі N_1 периодты, басқасы N_2 периодты. i -ші тізбегінің ($i = 1, 2$) мүшесін y_{ij} j -ші арқылы ($j = 1, 2, \dots$) және формула бойынша түзілетін жаңа тізбек мүшесін Z_j — j -ші арқылы белгілейік

$$z_j = (y_{1j} + y_{2j}) \bmod m. \quad (2.8)$$

Егер N_1 және N_2 периодтары—өзара қарапайым сандар болса, онда нәтижесінде шығатын Z_j тізбегі $-N = N_1 N_2$ периодымен мезгілді.

Процедураны (2.8) бастапқы тізбек санының арту жолымен жалпылауға болады. Мысалы, егер бастапқы тізбектің алғашқы саны g тең болса, онда нәтижесінде шығатын j -ші мүше формула бойынша алынады

арасындағы келісім, әдетте, χ^2 критеріінің көмегімен зерттелінеді.

3.8.1. Покер-тест

Бұл тест шығарылатын тізбек санында санның пайда болу кездейсоқтығын анықтай үшін арнайы құрастырылған. x_1, x_2, \dots, x_N $F(x)$ таралу функциясы бар кездейсоқ шама тізбегі делік. Кездейсоқ шама X мәнінің ауданын нүктелер $a_0 < a_1 < \dots < a_{k-1} < a_k$ көмегімен ктең ықтимал интервалға бөлейік. Егер X шынымен де $F(x)$ таралу

$$P(a_{j-1} \leq X < a_j) = \frac{1}{k}.$$

функциясы болатын болса, онда әр интервал үшін ($a_{j-1} \leq X < a_j$) болады $x_i \in [a_j; a_{j+1})$, $j = 0, 1, \dots, k-1$ үшін $y_i = j$ пайдаланып, бірдей ықтималдылықпен $0, 1, \dots, k-1$ мәнінің біреуін қабылдайтын жаңа y_i , $i = 1, 2, \dots$, кездейсоқ сан тізбегін аламыз.

Одан әрі барлық y_1, y_2, \dots тізбекті (y_1, y_2, \dots) (y_6, y_{10}, \dots) бесеуге болуге болады.

Әр түрлі цифр құрамымен ерекшеленетін бестіктің жеті класы бар:

- $abcde$ —бестіктегі барлық цифрлар әр түрлі;
- $aabcd$ —екі цифр бірдей, қалғандары әр түрлі;
- $aabbc$ —цифрдың екі жұбы бірдей;
- $aaabc$ —үш цифр бірдей, екеуі әр түрлі;
- $aaabb$ —үш цифр бірдей және екеуі бірдей;
- $aaaab$ —төрт цифр бірдей;
- $aaaaa$ —барлық цифрлар бірдей.

y_1, y_2, \dots тізбегіндегі цифрдың $0, 1, \dots, k-1$ әрқайсысы бірдей ықтималдылықпен пайда болады және осы тізбектің жеке мүшелері тәуелсіз деп болжап, әр түрлі $P(abcde)$, $P(aabcd)$, $P(aabbc)$, $P(aaabc)$, $P(aaabb)$, $P(aaaab)$, $P(aaaaa)$ бестіктің таралу ықтималдылығын анықтауға болады. Осы ықтималдылықтың мәні екі жиі қолданылатын $k=2, 8$ және 10 үшін 3.4 кестеде көрсетілген.

Бақыланатын таралумен әр түрлі типті бестіктің таралу келісімін χ^2 критеріінің көмегімен тексеріледі.


```

        if(y[j + i * 5] == y[m + i * 5])
            numN[j]++;
// j қайталанатын сан қанша рет кездескендігі
// сақталатын ауқымды нөльдеймізfor      (j
    = 0; j < 5; j++)    repN[j] = 0;
for (j = 0; j < 5; j++) repN[numN[j] - 1]++;
// бестік класын
анықтаймызif
(repN[0]==5) Nabcde++;
else
    if (repN[0]==3) Naabcd++;
    else
        if (repN[1]==4)
            Naabbc++; else
                if ((repN[1] == 2) && (repN[2] ==
                    3))
                    Naaabb++;
                else
                    if (repN[3]==4) Naaaab++;
                    else
                        if (repN[4]==5)
                            Naaaaa++; else Naaabc++;
}
// Әр түрлі класты бестіктің пайда болуы
//      (double)Nabcd
ЫҚТИМАЛДЫЛЫҒЫН e      / N5;
double Paabcd = (double)Naabc / N5;
double Paabbc = (double)Naabb / N5;
double Paaabc = (double)Naaab / N5;
double Paaabb = (double)Naaab / N5;
double Paaaab = (double)Naaaa / N5;
double Paaaaa = (double)Naaaa / N5;

```

3.8.1. Коллекция жиюшы критеріі

Тест өз коллекциясын толықтырушы коллекция жиюшы қызметіне ұқсас үрдісті іске асырады. Бұл критериді қолдану кезінде покер-тестті қолдану кезіндегідей түзілетін y_1, y_2, \dots тізбегі онда $0, 1, \dots, k-1$ санының барлық k пайда болғанға дейін ұзақ байқалады. R_1 – бақыланатын тізбек кесіндісінің ұзындығы делік. Осыдан соң тізбектің қалған бөлігі үшін «коллекциялау» үрдісі қайта қосылады, оның нәтижесінде R_2 тізбегінің жаңа ұзындығы алынады. ▭ Бұл


```

{
    r = random.NextDouble();
    x = Math.Log((1 + r) / (1 - r)) /
    k;
    r =
    random.NextDouble();
    if (r < 0.5) x = -x;
    Console.WriteLine("x[{0}] =
    {1:f3}", i + 1, x);
}
Console.WriteLine("Модельдеу аяқталды");
Console.ReadLine();

```

4.5.2. Орталық шекті теореманы пайдалану

Кездейсоқ шаманың қалыпты тараған бағдарламалық іске асырудың ең белгілі әдістерінің бірі орталық шекті теоремаға негізделген: тәуелсіз кездейсоқ шама соммасының таралуы n шексіз артуы кезінде қалыптыға жақындайды, егер төмендегі шарттар орындалса:

1) барлық шамалар соңғы математикалық күтілім мен дисперсияға ие;

2) мәні бойынша шаманың бірде біреуі басқа қалғандарынан қатты ерекшеленбейді.

Осы теоремаға сай интервалда $[0; 1)$ біркелкі тараған тәуелсіз кездейсоқ T_j, r_2, \dots, r_n шама соммасымен таралу жуықтауы $N(0,1)$ негізінде кездейсоқ X шаманы іске асыру алгоритмін құрастыруға болады. Олардың әрқайсысы математикалық күтілуге $M(r_i) = 1/2$ және дисперсияға $D(r_i) = 1/12, i = 1, 2, \dots, n$ ие болғандықтан, онда

$$M\left(\sum_{i=1}^n r_i\right) = \frac{n}{2}, \quad D\left(\sum_{i=1}^n r_i\right) = \frac{n}{12}.$$

Кездейсоқ шама таралуының орталық шекті теоремасына сай

$$x = \frac{\sum_{i=1}^n r_i - \frac{n}{2}}{\sqrt{\frac{n}{12}}} \quad (4.3)$$

n жеткілікті үлкен болған кезінде қалыптығыжақын.

қтайды. Арқырғы n таралу

Практика көрсеткендей, $n = 12$ болған кезде айтарлықтай жақсы


```

    Math.Cos(2 * Math.PI * r2); x
= m1 + x * Math.Sqrt(d1); y =
Math.Sqrt(-2 * Math.Log(r1)) *
    Math.Sin(2 * Math.PI * r2); y
= m2 + y * Math.Sqrt(d2);
Console.WriteLine("x[{0}] = {1:f3} \ty[{0}] =
{2:f3}", i + 1, x, y);
}
Console.WriteLine("Модельдеу аяқталды");
Console.ReadLine();
}

```

5.4.5. Марсаль мен Брей әдісі

Марсаль мен Брей әдісі Бокс пен Маллер әдісінің модификациясы болып табылады. Бұл әдіс кездейсоқ X және Y шамалары тәуелсіз болатын жағдайда бірлескен тығыздық функциясы $f(x, y) = f(x)f(y)$ берілген екі өлшемді кездейсоқ $[X, Y]$ шаманы іске асыруға арналған. Бұл әдіс есептелінген синустар мен косинустарды алып тастау арқасында жылдамырақ болып табылады. Кездейсоқ шаманы алу процедурасы келесіге негізделінген:

1) U_1 интервалда $[-1; 1)$ біркелкі тараған тәуелсіз, кездейсоқ U_1 және U_2 шамалар түрленеді:

$$U_1 = -1 + 2R_1;$$

$$U_2 = -1 + 2R_2,$$

мұндағы R_1, R_2 – интервалда $[0; 1)$ біркелкі тараған тәуелсіз, кездейсоқ сан;

2) $U_1^2 + U_2 < 1$ шарттары тексеріледі. Егер шарттар орындалмаса, онда циклді қайта қайталайды. Егер шар дұрыс болса, онда формула бойынша кездейсоқ шаманы іске асырады

$$X = U_1 \sqrt{\frac{-2 \ln(U_1^2 + U_2^2)}{U_1^2 + U_2^2}}, Y = X \frac{U_2}{U_1}.$$

Әдіс жақсы нәтижелер береді және оңай бағдарламаланады. Сонымен қатар осы әдіс келесі кемшіліктерге ие:

- айтарлықтай баяу, өйткені онда ұзақ есептілінетін тригонометриялық функциялар қолданылады;
- нөлге жақын U_1 , кезінде ЭЕМ қатарлық торының шамадан тыс толуы мүмкін.

Осы кемшіліктер кездейсоқ шаманың миллионын алуды талап ететін стохастикалық модельдерде генераторды пайдаланған жағдайда елеулі болып табылады.

4.6. АРНАЙЫ ТАРАЛУ ЗАҢЫ БАР КЕЗДЕЙСОҚ ШАМАНЫ ТҮРЛЕНДІРУ

Қосымшада ең жиі кездесетін кездейсоқ шама таралуының бірқатар заңдарында модельдеудің жиі тәсілдері әзірленген, жиі тәсілдер жалпы әдістерді қолдану мүмкіндігін алып тастамайды. Әр нақты тапсырманың шешімі модельдеу алгоритмін таңдауда жеке тәсілді талап етеді. Көрсеткіштердің бірі біркелкі таралған кездейсоқ сан үшін қолданатын санына таралуын талап ететін жалған кездейсоқ шама мөлшеріне қатысты түсінік деп қабылдайтын *әдістің тиімділігі* болып табылады.

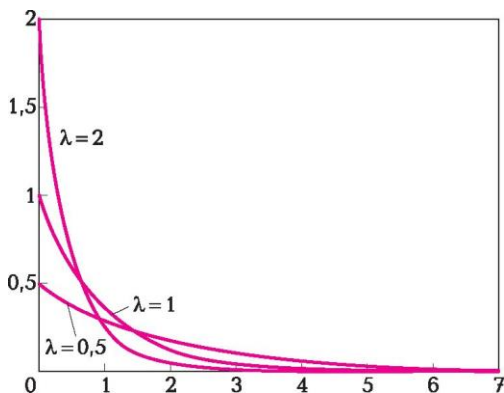
4.6.1. Экспоненциалды таралуды модельдеу

Экспоненциалды немесе көрсеткішті таралу – бұл бірдей уақиғаның екі тізбекті жасауы арасындағы уақытты, құрылғының мүлтіксіз жұмыс істеу уақытын модельдейтін үздіксіз тарату.

Экспоненциалды таралу тығыздығының функциясы теңдеумен беріледі

$$f(x) = \lambda e^{-\lambda x}, \quad x \geq 0,$$

мұндағы X – таралу көрсеткіші, $\lambda > 0$.



Сур. 4.8. Экспоненциалды таралу тығыздығының функциясы

Экспоненциалды таралу үшін маткүтілуі мен дисперсия теңдеуге сай анықталады

$$M(X) = \lambda;$$

$$D(X) = \lambda^2.$$

Ықтималдылық тығыздығы функциясының графигінің әр түрлі мән үшін X көрсеткіші 4.8. суретте көрсетілген.

X көрсеткіші бар экспоненциалды таралудың кездейсоқ шамасын модельдеуге арналған формула төмендегі түрге

$$x = -\frac{1}{\lambda} \ln r.$$

Модельдеу мысалы 4.1 тармақшада қарастырылды.

4.6.2. Бета-таралуды модельдеу

Бета-таралу ақырғы жоғарғы және төменгі шекті мәндермен үздіксіз таралу болып табылады. Осындай шектеулер көптеген нағыз уақиғаларда болуы мүмкін, сондықтан бета-таралуды жеткілікті ақпаратты үлкен мөлшерде алуға дейін нағыз таралуды жуықтау үшін қолдануға болады. Бета-таралу жеткілікті шамада көптеген деректерге сай келеді.

Интервалдағы $(0; 1)$ бете-таралудың тығыздығы формуламен анықталады

$$f_{p,m}(x) = \frac{x^{p-1}(1-x)^{m-1}}{B(p,m)}, \quad 0 < x < 1,$$

мұндағы $p > 0$, $m > 0$ —таралу көрсеткіштері.

Бета-функция төмендегі түрге ие

$$B(p,m) = \int_0^1 x^{p-1}(1-x)^{m-1} dx.$$

Бета-таралу үшін маткүтілу мен дисперсия формулаға сәйкес анықталады.

4.6.5. Вейбулл таралуын Модельдеу

Вейбулл таралуы көмегімен қандай да бір тапсырманы орындау уақыты, шамасы, құрылғының дамылсыз жұмыс істеу уақыты жиі сипатталынады.

Вейбулл таралуына арналған тығыздық функциясы келесі түрге ие

$$f(x) = \frac{cx^{c-1}}{b^c} \exp\left(-\left(\frac{x}{b}\right)^c\right),$$

мұндағы $0 \leq x < \infty$; $c > 0$; $b > 0$; b, c – бүтін.

Вейбулл таралуы үшін маткүтілу мен диспесия теңдеуге сай анықталады

$$M(X) = \frac{b}{c} \Gamma\left(\frac{1}{c}\right);$$

$$D(X) = \frac{b^2}{c} \left[2\Gamma\left(\frac{2}{c}\right) - \frac{1}{c} \left[\Gamma\left(\frac{1}{c}\right) \right]^2 \right].$$

Мұнда $\Gamma(\cdot)$ – гамма функция.

$b = 1$ мен c әр түрлі мәнінің көрсеткіштері үшін ықтималдылық тығыздығы функциясының графигі 4.12 суретте көрсетілген.

Вейбулл таралуы келесі формуланы қолдану жолымен модельденуі мүмкін:

$$x = b[-\ln(r)]^{1/c},$$

мұндағы r – интервалда $(0; 1)$ біркелкі тараған кездейсоқ шама. Модельдеу кезінде r шамасы төменгі жағынан кіші шамамен шектелуі керектігін ескерейік.

Мысал. Вейбулл таралуына жауап беретін кездейсоқ X шаманы модельдеу бағдарламасын құрастыру.

Шешімі. C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
{
    Console.WriteLine("Тарату көрсеткіштері");
    Console.Write("c: ");
    double c = double.Parse(Console.ReadLine());
    Console.Write("b: ");
    double b = double.Parse(Console.ReadLine());
```

Егер зерттелінетін ЖҚЖ үшін оның сипаттамаларын бағалауға арналған аналитикалық көріністі алу мүмкін болмаған немесе берілген көрініс практикада қолдану үшін тым күрделі болып шыққан жағдайда, ЖҚЖ тиімділігін компьютерлік модельдеу көмегімен зерттеуді орындауға болады. ЖҚЖ модельдеудің әр түрлі тәсілдерінің нақты мысалдары [2] қарастырылған.

ЖҚЖ модельдеу үшін алгоритмдерді құру әдісінің келесі негізгі әдістері қолданылуы мүмкін [2]:

- ауыспалы қадамы бар мезгілді модельдеуі (немесе модельді уақыттың уақиғадан уақиғаға қозғалу әдісі);
- тұрақты қадамы бар мезгілді модельдеу;
- өтініштің тізбекті өткізуі;
- өтініштің сатылы тізбекті өткізілуі.

Берілген алгоритмдер бір мәннен басқасына модельді уақыт сағатында сақталатын модельді уақыттың қозғалу әдісімен ерекшелінеді. Бірінші екі алгоритмдердің мәнін қысқаша қарастырайық, одан кейін көбіне ЖҚЖ модельдеу кезінде қолданылатын өтініштің тізбекті өткізілу әдісін егжей-тегжейлі сипаттайық.

Уақыттың уақиғадан уақиғаға қозғалуын пайдалану кезінде бастапқы күйдегі модельді уақыттың сағаты нөлге орнатылады және жүйе күйінің өзгеруіне әкеп соғатын болашақ уақиғаның пайда болу уақыты анықталынады [5]. Осыдан соң модельді уақыт сағаты жақын уақиғаның пайда болу уақытына өтеді және сол мезетке өткен уақиғаны ескере отырып жүйе күйі, сонымен қатар болашақ уақиғаның пайда болу уақыты жайлы мәлімет жаңаланады. Кейін модельді уақыт сағаты келесі жақын уақиғаның пайда болу уақытына жылжиды, жүйе күйі жаңаланады және болашақ уақиға уақыты анықталынады және с.с. модельді уақыттың бір уақиғаның пайда болу уақытынан басқасының пайда болу уақытына жылжу процесі алдын-ала көрсетілген қандай да бір тоқтату шарты орындалмағанға дейін жалғасады. Осындай имитациялық модельде барлық өзгерістер тек уақиға пайда болған кезде ғана орын алатындықтан, жүйенің әрекетсіздік периоды жай өткізіп жіберіледі және сағат бір уақиғаның пайда болу уақытынан басқасының пайда болу уақытына ауыстырылады.

Тұрақты қадамы бар модельді уақыттың жылжуы кезінде модельді уақыт сағаты нақты Δt мәніне арналған уақыттың дәл Δt бірлігіне жылжиды. Сағаттың әр жаңаруынан кейін алдыңғы уақыт Δt интервалы

аралығында қандай да бір уақиға жүргенін анықтау мақсатымен тексеру орындалады. Егер осы интервалға бір немесе бірнеше уақиға жоспарланған болса, берілген уақиға интервалдың аяғында жүрді деп есептелінеді, содан кейін жүйенің күйі мен статистикалық есептегіштер сәйкесінше жаңарады.

Тұрақты қадам арқасында уақыттың жылжуы екі кемшілікке ие:

- уақиғаның жүру аралығында интервал соңындағы уақиғаны өңдеумен байланысты қатенің пайда болуы;
- егер шынымен де әр түрлі уақытта жүретін уақиға бір уақыттық деп қарастырылса, қандай уақиғаны бірінші өңдеуді шешу қажеттілігі.

Осындай мәселелерді Δt интервалын ұзақтығы азырақ қылып жартылай шешуге болады, бірақ онда уақиғаның пайда болуын тексеру саны артады, ол тапсырманың орындалу уақытының көбеюіне әкеліп соғады. Осыдан ЖҚЖ модельдеуде тізбекті уақиғалар арасындағы уақыт интервалдары ұзақтығы бойынша біршама ерекшеленетін кезде, тұрақты қадам көмегімен уақыттың жылжуы қолданылмайды.

Бірақтар жағдайда ЖҚЖ өтінішке қызмет көрсету процестерін модельдеу кезінде өтінішті тізбекті өткізу әдісі бойынша модельдеуші алгоритм құрған ыңғайлы [2]. Бұл әдіс жүйеде жүріп жатқан уақиғалар бір біріне тәуелді болмаған кезде қолданылуы мүмкін. Модельдеуші алгоритм жеке өтініштердің тарихын олардың жүйегі түсу тәртібінде бірізді қайта жаңғыртады. Алгоритм басқа өтініштер жайлы ақпаратқа берілген өтінішке одан әрі тәртіпте қызмет көрсету жайлы мәселені шешу үшін қажет болса ғана жүгінеді.

Тізбекті өткізілім әдісінің екі түрі бар:

- жалғыз өтінішті өткізу;
- өтініш ағысын өткізу.

Берілген әдіс өтініштің қалыптасу және қызмет көрсету ретімен ғана ерекшеленеді. Бірінші күтілімі бар бір каналды ЖҚЖ модельдеуге қолданылатын жалғыз өтінішті өткізу [2] әдісін қарастырайық. Модельдеу процесінде кезекті өтінішті қалыптастыру (бұл үшін өтініштің келу уақытының арасындағы уақыт интервалын анықтайтын кездейсоқ сан мен оған қызмет көрсетудің кездейсоқ уақыты түрлендіріледі) және қызмет көрсету уақытының басталуы мен аяқталуын анықтап, оның қызмет көрсетуін орындау қажет.

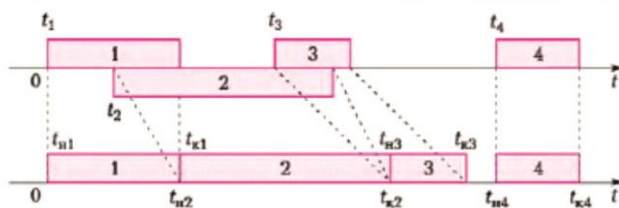
Модельдеудің басында модельді уақыт сағаты нөлге орнатылады. Бірінші өтініш жүйеге $t_1 = A_1$ уақыт мезетінде түсті делік. Мұнда A_1 – өтініштің түсуі арасындағы уақыт болып табылатын кездейсоқ

шаманы іске асыру. Уақыт t_1 мезетінде қызмет көрсету каналы бос болғандықтан, өтінішке қызмет көрсету уақыт $t_{k1} = t_1$ мезетінде басталады. Кейін бірінші S_1 өтінішке қызмет көрсету уақытының кездейсоқ шамасын алу керек. Сонда 1-ші өтінішке қызмет көрсету мезетінің аяқталуы $t_{k1} = t_{r1} + S_j$. Сонымен қатар қызмет көрсетілген өтініштің есептегіш санына бірлік қосылады.

Одан әрі келесі 2-ші өтініштің түсу уақыты анықталады. Ол үшін өтініштің A_2 түсуі арасындағы уақыт қалыптастырылады және 2-ші өтініштің түсу уақытының мезетін анықталынады $t_2 = t_1 + A_2$. Егер берілген мезетте, мысалы канал бос болмаса, онда 2-ші өтініш үшін ұзақтығы $\Delta T_{кҮТ} = t_{k1} - t_2$ күту периоды басталады. Бұл өтінішке қызмет көрсету $D_2 = t_{r1}$ мезетінде басталады. Одан әрі 2-ші өтінішке қызмет көрсету уақытының кездейсоқ шамасын S_2 анықталынады. Онда 2-ші өтінішке қызмет көрсетудің аяқталу мезеті $t_{k2} = t_{n2} + S_2$. Қызмет көрсетілген өтініш есептегішінің санына бірлік қосылады. Кейінгі өтініштерге ұқсас тәсілмен қызмет көрсетіледі.

Төрт өтінішті қалыптастыру және оған қызмет көрсету процесі 7.4 суретте көрсетілген. Жоғарғы уақыттық осі өтініштің келу уақыты (тік бұрыштың сол шекарасы) мен қызмет көрсетілуін (тік бұрыштың ұзындығы), ал төменгісі – өтінішке қызмет көрсету уақытының басталуы (тік бұрыштың сол шекарасы) мен аяқталуын (тік бұрыштың оң шекарасы) көрсетеді.

Өтініштің тізбекті өткізілім әдісінің екінші нұсқасы бірінші барлық өтініштің ағымы қалыптасатын, ал кейін өтінішке қызмет көрсету процесі басталатын әдіс болып табылады [2]. Сонымен



Сур. 7.4. Өтініштің бірізді өткізілім әдісінің көрінісі

қатар барлық өтініштер біркелкі деп болжанады. Бұл жағдайда өтініш жайлы деректерді ЭЕМ жадында сақтау үшін сандардың ауқымын құру керек. Осындай ауқымның мөлшері алдын ала белгісіз болғандықтан, оны белгілі-бір қормен таңдауға тура келеді,

ол жады босқа шығындауға әкеп соғады.

Одан әрі өтініштің бірізді өткізу әдісінің екінші нұсқасы қолданылатын, шексіз күтілімі бар бір каналды ЖҚЖ C# тілінде модельдеу бағдарламасының мәтінінің кесіндісі көрсетілген.

Кіретін өтініш ағысын қалыптастыру

```
: double time = 0; // Модельді уақытты нөлдеу
int i, Nin = 0; // Түсетін өтініш есептегішін нөлдеу
// кіретін өтінішті қалыптастыру
for (i= 0; i<Zmax; i++)
{
    // Кезекті өтініштің түсу уақытын қалыптастыру time=
    time+ rndexpo(lambda);
    // Егер модельдеу уақыты аяқталмаса if
    (time<modelTime)
    {
        // онда түсетін өтініш санын арттыру Nin++;
        // өтініштің келу уақытын есте сақтау zarrive[i]
        = time;
        // өтінішке қызмет көрсету ұзақтығын
        қалыптастыру
        zserve[i] = rndexpo(mu);
    }
    else break; // модельдеу уақыты аяқталу кезіндегі
    шығу
```

Бағдарламада қолданылатын тұрақты келесі тағайындауға ие:

λ —кіретін өтініш ағымының қарқындылығы;

μ —өтінішке қызмет көрсету ағымының қарқындылығы;

Z_{max} —қызмет көрсетілген өтініштің максималды саны (өтініштің келу уақыты мен қызмет көрсетілуі жайлы ақпарат сақталған z_arrive және z_serve ауқымының мөлшерін анықтайды);

$modelTime$ —модельдеу уақыты.

Бағдарламаның ұсынылған кесіндісінде функция дәлелімен анықталынған параметрі бар экспоненциалды таралған кездейсоқ шаманы түрлендіретін `rnd_expo` пайдаланушы функциясы қолданылған.

Одан әрі өтініш өткізілу бағдарламасының кесіндісін келтіреміз.

```
time= 0; // Модельді уақытты нөлдеу
intNserve= 0; // Қызмет көрсетілген есептегіш
санын
//нөлдеу
//барлық түскен өтінішті сұрыптауfor
(i= 0; i<Nin; i++)
{
    // Ағымдағы өтінішке қызмет көрсету басталуын
    анықтайif (z arrive[i] <time) timeServe = time;
    else timeServe = z arrive[i];
    //егер модельдеу уақыты аяқталмасаif
    (timeServe<modelTime)
    {
        // онда өтінішке қызмет көрсету уақытының
        басталуын
        //есте сақтау
        z begin[i] = timeServe;
        time = timeServe + z serve[i];
        //қызмет көрсету уақытының аяқталуын есте
        сақтайzend[i] = time;
        //қызмет көрсетілген өтініш санын
        арттыруNserve++;
    }
    elsebreak; //модельді уақыттың аяқталуы кезіндегі
    шығу
}
// соңғы өтінішке қызмет көрсету уақытының
аяқталуыdoubletimeEnd= time;
```

Бағдарламаны әзірлеу кезінде, моделді уақыттың аяқталуына дейін қызмет көрсетіліп бастаған өтінішке міндетті түрде қызмет көрсетілетіндігі болжанғанын ескерейік. Одан кейінгі өтініштерге қызмет көрсетілмейді.

ЖҚЖ қызмет ету сипаттамаларын есептеу мен экранға шығаруға арналған бағдарлама кесіндісін елестетейік.

```
double timeWait = 0; // өтініштің кезекте күту
уақыты
double timeSys = 0;// өтініштің жүйеде болу уақыты
double timeWork = 0; // жүйенің жұмыс істеу уақыты
//қызмет көрсетілген
өтініштерді сұрыптауfor (i =
0; i <Nserve; i++)
{
```

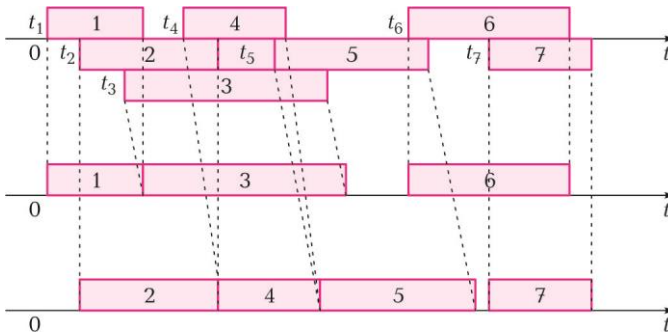
```

//кезекте күту уақытын
жинақтауtimeWait += (z begin[i] - z
arrive[i]);
// жүйеде болу уақытын жинақтауtimeSys
+= (zend[i] - zarrive[i]);
// жүйенің жұмыс істеу уақытын
жинақтауtimeWork += zserve[i];
}
Console.WriteLine("Түскен өтініш саны:{0}", Nin);
Console.WriteLine("Қызмет көрсетілген өтініш
саны:{0}", Nserve);
Console.WriteLine("Кезекте күткен орташа
уақыт:{0^4}", (double) timeWait/ Nserve);
Console.WriteLine("Жүйеде болған орташа уақыт:
{0:f4}", (double) timeSys/ Nserve);
Console.WriteLine("ЖҚЖ қолдану коэффициенті:
{0:f4}", (double) timeWork/timeEnd);

```

Өтінішті бірізді өткізудің қарастырылған әдісін көп каналды ЖҚЖ модельдерін құру үшін де қолдануға болады.

Мысал үшін 7.5 суретте екі каналды ЖҚЖ жұмысын модельдеу процесі көрсетілген. Суретте көрсетілгендей, берілген жағдайда



Сур. 7.5. Екі каналды ЖҚЖ модельдеуге арналған өтініштің бірізді өткізу әдісінің көрінісі

Өтініштің кіретін ағымы (жоғарғы уақыттық ось) қызмет көрсетудің екі құрылғысына (төменгі уақыттық ось) таралады. Әрі, өтініштің келген мезетінде бос құрылғылар бар болса, онда оған олардың кез-келгенімен қызмет көрсетіледі (әдетте бірінші босы

таңдалынады). Егер барлық қызмет көрсету құрылғылары бос болмаса, онда өтініш ерте босайын құрылғыға бағытталады.

Одан әрі servCount канал санымен күтілетін көп каналды ЖҚЖ өтінішке қызмет көрсету және сипаттамаларды анықтау бағдарламасының кесіндісі ұсынылған(бұл кезде өтінішті қалыптастыру бір каналды ЖҚЖ өтінішті қалыптастыруға толығымен сәйкес келеді).

```
// басында барлық қызмет көрсететін
құралдар бос for (j = 0; j <servCount; j++)
endTime[j] = 0;
int Nserve = 0; // Қызмет көрсетілген өтінішті
есептегіш
//санын нельдеу
double timeWait = 0; // өтініштің кезекте күту
уақыты
double timeSys = 0; // өтініштің жүйеде болу
уақыты
double timeWork = 0; // жүйенің жұмыс істеу
уақыты
//түскен өтініштерді
сұрыптауfor (i= 0; i<Nin;
i++)
{
    //Ерте босайтын ind каналының нөмірін
анықтаймыз
    ind = 0;
    for (j = 1; j <servCount; j++)
        if (endTime[j] < endTime[ind])
            ind = j;
    // егер барлық қызмет көрсету
құралдары бос //болмаса қызмет
көрсетудің басталу уақытын
// анықтайif (z arrive[i] <
endTime[ind]) timeServe =
endTime[ind]; else
    timeServe = z arrive[i];
    // егер уақыт аяқталмасаif
(timeServe < modelTime)
    {
// онда өтінішке қызмет көрсетудің басталу уақытын
есте сақтауз
        zbegin[i] = timeServe;
        // қызмет көрсетудің аяқталу уақытын есте
```

```

сақтау zzend[i] = timeServe + zserve[i];
// жүйедегі жұмыс уақытын жинақтауtimeWork
+= z serve[i]; endTime[ind] = z end[i];
// қызмет көрсетілген өтініш санын
арттыруNserve++;
// кезектегі күту уақытын
жинақтауtimeWait+= (zbegin[i] -
zarrive[i]);
// жүйеде болу уақытын жинақтауtimeSys +=
(z end[i] - z arrive[i]); if (timeEnd <
endTime[ind]) timeEnd = endTime[ind];
}
elsebreak; //модельдеуді аяқтау
}
Console.WriteLine("Түскен өтініш саны:{0}", Nin);
Console.WriteLine("Қызмет көрсетілген өтініш
саны:{0}", Nserve);
Console.WriteLine("Кезекте күткен орташа
уақыт:{0^4}", (double) timeWait/ Nserve);
Console.WriteLine("Жүйеде болған орташа уақыт:
{0:f4}", (double) timeSys/ Nserve);
Console.WriteLine("ЖҚЖ қолдану коэффициенті:
{0:f4}", (double) timeWork/timeEnd/servCount);

```

ЖҚЖ соңғы талқыланатын модельдеу әдісі – өтінішті сатылы бірізді өткізу әдісі, әр түрлі басымдылықғы бар ЖҚЖ модельдеуге арналған. Берілген әдіс өтініштің бірізді өтілімі әдісіне негізделген және бірінші басымдылығы жоғарырақ өтінішке қызмет көрсету процесі, ал кейін басымдылығы төмендеу өтінішке қызмет көрсету процесі модельденеді. Сонымен қатар қықмет көрсету құрылғысының жұмыс уақытының бөлігі басымдылығы жоғарырақ өтініштерге қызмет көрсетумен бос емес болатындығы ескеріледі. Басқаша айтқанда, басымдылығы төмендеу өтініштерге қызмет көрсету құрылғысы бос тұрған уақытта ғана қызмет көрсетілкі керек. Сонымен бірге абсолютті басымдылық іске асырылады, онда басымдылығы төмендеу өтінішке қызмет көрсету мезетінде келген басымдылығы жоғарылау өтініш басымдылығы төмендеу өтінішті ығыстырады. Берілген өтініш қызмет көрсету құрылғысы босаған соң қызмет алуын жалғастырады.

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

1. ЖҚЖ деген не? Осындай жүйеге мысал келтір.
2. ЖҚЖ негізгі компоненті қандай? Олар қандай параметрлерді береді?
3. ЖҚЖ жіктеудің негізгі әдістері қандай? ЖҚЖ жіктеу беріңіз.
4. ЖҚЖ қандай негізгі сипаттамаларын білесіз?
5. Бір каналды ЖҚЖ деп нені түсінеді? Оны суреттеңіз.
6. Бір каналды ЖҚЖ сипаттамаларын қалай аналитикалық есептейді?
7. Көп каналды ЖҚЖ қалай түсінесіз? Оны суреттеңіз.
8. Көп каналды ЖҚЖ қалай аналитикалық есептеуге болады?
9. Қандай ЖҚЖ жабық деп аталады?
10. Уақиғадан уақиғаға модельді уақытты жылжытып модельдеу әдісі неге негізделген?
11. Тұрақты қадамы бар модельді уақытты жылжытып модельдеу әдісі неден тұрады?
12. Өтініштің бірізді өтілімі әдісі неге негізделген?
13. Өтініштің сатылы бірізді өткізілім әдісі неден тұрады?

КОМПЬЮТЕРЛІК ЭКСПЕРИМЕНТТІ ЖОСПАРЛАУ

8.1. НЕГІЗГІ ҰҒЫМДАР

Эксперименттің мақсаты белгіленгенін және машиналық имитация моделін қолдану шешімі қабылданғыннан кейін, алдағы эксперименттің бастапқы жоспарлауын жүргізу тиімді болып саналады.

Қажетті ақпаратты нысаналы және тиімді алу үшін зерттеушіде толық жоспар болу керек. Уақыт пен шығынға арналған жоспарлық шектеулер зерттеушінің құзырындағы ресурстарға сәйкес келуі қажет. Эксперимент неғұрлым қиын әрі қымбат болса, сол кезеңге соғұрлым назар аудару қажет.

Эксперимент жоспары төмендегілерді анықтайды:

- Компьютердегі есептеулер көлем;
- Компьютердегі есептеулерді жүргізу реті порядок;
- Жиынтық тәсілдері және үлгілеу нәтижелерін статистикалық өңдеу.

Эксперименттерді жоспарлау келесі мақсаттардан құралады:

- Нәтижелердің дәлдігі мен сенімділігіне көзделген талаптарды жүргізу барысында үлгілеудің ортақ уақытын қысқарту;
- әрбір бақылаудың ақпараттылығын арттыру;
- зерттеу процесінің құрылымдық негізін құру. Эксперименттерді жоспарлау теориясының негізгі ұғымдарын құрастырамыз, толығырақ [12 қосымшада. Экспериментті жоспарлау кезіндегі айрықша келісімді үлгі «қара жәшік» (8.1 сурет) деп аталатын абстрактылы сызба болып табылады.
- Сондай тәсілдеме кезінде ішкі және сыртқы айнымалалар ажыратылады: $x_1, x_2, \dots, x_k; y_1, y_2, \dots, y_m$. Кез келген айнымалы не факто, не реакция болып табылады.



8.1 сур «Қара жәшік» күйіндегі жүйе үлгісі

Мысалы, тек екі түрлі айнымалы бар делік. Олар: x пен y . Егер эксперименттің мақсаты x айнымалысының y айнымалысына әсерін зерттеу болса, онда x — фактор ал y — реакция болып табылады.

Әрбір фактор x_i , $i = 1 \dots k$ экспериментте бірнеше айнымалылардың арасынан *деңгей* деп аталатын бір мәнге ие бола алады. Барлық факторлардың белгіленген деңгейлер жиынтығы қарастырылып жатқан жүйенің мүмкін жағдайларының біреуін анықтайды.

Факторлардың әрбір белгіленген деңгейлер жиынтығы *факторлық кеңістік* деп аталатын көпөлшемді кеңістіктегі анықталған нүктеге сәйкес келеді.

Байланыс күйінде өрнектеуге болатын факторлар деңгейлері мен жүйелер реакциясының арасындағы толақтай анықталған байланыс бар.

$$\tilde{y}_j = \varphi_j(x_1, x_2, \dots, x_k), \quad j = \overline{1, m}.$$

Факторлармен реакцияларды байланыстыратын функцияна реакция функциясы деп атайды, ал оған сәйкес геометриялық бейне реакция жазықтығы деп аталады. Реакция функциясы жайлы хабардар болу зерттеушіге жүйенің жұмысы туралы толық ақпарат береді. Ол жүйедегі қызықтыратын сипаттамаларды басқаруды, белгіленген тиімділік өлшемдеріне сәйкес факторлардың тиімді мәндерін жинақтауды оңай үйрене алады. Сондықтан зерттеушіге үлгілеу кезінде реакция қандай қалыпта факторға бағынышты болатындығын білу маңызды.

Сонымен қатар зерттеушіге y , $j = \overline{1, m}$ тәуелділік түрлері адын ала белгісіз, сол себепті жуық байланыс қолданылады:

$$\tilde{y}_j = \varphi_j(x_1, x_2, \dots, x_k), \quad \varphi = \overline{1, m}.$$

Әдетте j тәуелділігі ретінде бірінше және екінші дәрежелі полиномдар сайланады. Олардың коэффициенттері эксперименттің мағлұматтары бойынша (мысалы, ең кіші квадраттар тәсілінің көмегімен) табылады. Коэффициенттерді есептеленетін ресустарының минималды шығыны (мысалы, сынақтардың

минималды саны) кезінде ішкі айнымалыдардың мәндер ережелерін арнайы тұдырымдаулары бойынша түрлендіре математикалық үлгіні құрастырып, оның сипаттамаларын бақылайтындай етіп орнату қажет.

Эксперименттерді жоспарлау кезінде факторлардың негізгі ерекшеліктерін анықтау қажет. Эксперименттерді жүргізу кезіндегі факторлар басқарылатын және басқарылмайтын, зерттелетін және зерттемейтін, бақыланатын және бақыланбайтын, есептік және сапалық, белгіленген және кездейсоқ болуы мүмкін.

Фактор басқарылатын деп аталады, егер оның деңгейлері эксперимент кезінде зерттеушімен нысаналы түрде анықталған жағдайда. Фактор бақыланатын деп аталады, егер оның мәндері бақыланады және тіркеледі. Фактор зерттелетінге жатады, егер ол қосымша нысана (мысалы, эксперименттің дәлдігін арттыру үшін) ретінде емес жүйенің ерекшеліктерін зерттеуге арналған үлгіге кірген жағдайда. Фактор есептік болып табылады, егер оның мәні— реакцияға әсер ететін сандық шамалар, қарама-қайшы жағдайда фактор сапалық болып есептелінеді. Мысалы, БҚКЖ есептік факторлар деп сұрыныстардың қарқынды кіріс ағыны, қызмет көрсету ағынының қарқыны, жинақтауыш сыйымдылығы, қызмет көрсететін арналардың саны және т.б. есептелінеді, ал сапалық факторлар— тізімге тәртіп орнатулар, тізімнен талғаулар, сұраныстарға ағындармен қызмет көрсету және т.б.

Фактор белгіленген деп аталады, егер экспериментте экспериментаторды қызықтыратын факторлардың барлық мәндері зерттелетін болса, ал егер экспериментаторды қызықтыратын факторлар мәндерінің қосындысынан тек кейбір кездейсоқ сұрыптауларды зерттейтін болса онда фактор кездейсоқ деп аталады. Кездейсоқ факторлар негізінде болжамалы тұжырымдар және экспериментте зерттелмеген фактор мәндері жасалынуы мүмкін.

Факторға қойылатын негізгі талаптар факторды басқару талаптар, нысанға тура әсер ету талаптары, факторлардың үйлесімділігі мен тәуелсіздігі талаптары болып табылады.

Фактордың басқарымдылығы деп барық сынақ аралығында таңдалған керекті фактор деңгейін тұрақты орнату және ұстап тұру немесе берілген бағдарламаға сай өзгертін мүмкіндік деп түсінеді. Нысанға тура әсер ету талаптары үлкен маңыздылыққа ие, өйткені фактор басқа факторлардың функциясы болып табылса, оны басқару қиын. Факторлардың үйлесімділігі олардың барлық қиыстыруларын іске асыруға болатындығын, ал тәуелсіздік басқалардың деңгейіне тәуелсіз кез-келген деңгейде факторды орнату мүмкіндігіне сай

келетіндігін білдіреді.

Машиналық тәжірибелерді жоспарлау кезінде екі құраушыны бөлуге болады: стратегиялық және тактикалық жоспарлау.

Стратегиялық жоспарлаутәжірибе қоюшының қарамағында бар қордың шектеуін ескере отырып, ЭЕМ іске асырылған модель көмегімен жүйе жайлы қажетті ақпаратты алу тапсырмасын шешуді өзінің негізгі мақсатына қояды. Машиналық тәжірибені жоспарлау кезіндегі жиі тапсырмалардың санына модельдеуге кететін машина уақытын азайту, модельдеу нәтижесінің дәлділігі мен нақтылығын арттыру модельдің барабарлығын тексеру міндеттері жатады.

Тактикалық жоспарлаутәжірибе жоспарында қарастырылған машина модельінің әр сынақ топтамасын жүргізу әдісін анықтауды көрсетеді.

8.2.

ТӘЖІРИБЕНІ СТРАТЕГИЯЛЫҚ ЖОСПАРЛАУ

Модельдеудің стратегиялық жоспарын құру процесі негізгі екі сатыдан тұрады: тәжірибенің құрылымдық моделін құру; функционалды модельді құру.

Құрылымдық модельді құру кезінде факторлар саны мен әр факторға арналған деңгей саны таңдалады. Осы көрсеткіштерді таңдау тәжірибенің мақсатына, фактор өлшемдерінің дәлділігіне, сызықсыз эффектке қызығушылық және с.с. анықталынады. Осы таңдауға қордың шектелгендігі арқасында орын алатын мүмкін өлшеу сандарының шектелгендігі әсер етпеуі керек, яғни құрылымдық модель не істеу керектігіне сүйеніп таңдалынады.

Құрылымдық модель, сәйкесінше, түрге ие

$$N_s = q_1 q_2 \dots q_k,$$

Мұндағы N_s —тәжірибе элементтерінің саны; k —тәжірибе факторларының саны; q_i — i -ші фактор дәрежесінің саны.

Мұнда элемент ретінде фактордың нақты мән жиынында жүргізілетін қарапайым тәжірибе деп түсінеді.

Қажетті фактордың саны мен түрін таңдау кезінде төмендегідей мәселелер бірізді шешіледі:

1) бізді бірінші кезекте қызықтыратын қандай жауапты (шығатын ауыспалы) анықтайды;

2) осы ауыспалыларға қандай факторлар әсер ететіндігін

белгілейді. Жүйенің көпшілігі Парето қағидасына сай жұмыс істейтіндіктен, онда ең маңызды факторлардың 20 %жүйенің қасиетінің 80 %анықтайды, ал фактордың қалған 80 % оның қасиетінің тек 20 % анықтайды;

3)өлшенетін және бақарылатын факторларды бөледі және олардың мүмкін деңгейін анықтайды. Әр фактор үшін деңгей санын минимальды мүмкін, бірақ тәжірибе мақсатына жету үшін жеткілікті таңдау керек. Барлық факторлар үшін бірдей q санының бірдей қашықтықтағы деңгейі (екі-үш) модель зерттеуін айтарлықтай жеңілдетеді. Бұл жағдайда құрылымдық модель төмендегі түрге ие

$$N_s = q^k.$$

Функционалды модельжауаптың нағыз өлшеміне қызмет ететін құрылымдық модель элементтерінің санын қамтамасыз етеді, яғни әр түрлі ақпараттық нүктенің қанша болуы қажеттігін анықтайды. Ұқсас функционалды модельдер не аяқталған не аяқталмаған болуы мүмкін. Егер жауап өлшемінде оның барлық элементтері қатысса, функционалды модель аяқталған деп аталады, яғни $N_f = N_s$. Егер орын алған жауап саны элемент санынан аз болса, функционалды модель аяқталмаған деп аталады, яғни $N_f < N_s$. Мінсіз жағдайда функционалды модель құрылымдыққа сай келеді. Бірақ модельді зерттеулердің көпшілігі уақытқа, ақшалай қаражатқа және есептегіш машиналардың өндірімділігіне негізделген шектеуге ие болғандықтан, функционалды модельді құру кезінде қалау мен ресурс арасында лайықты ымыраны таңдау керек. Функционалды модельді құру кезіндегі одан кейінгі әрекет жүргізілетін зерттеудің мақсатына тәуелді. Мысалы, реакция бетін зерттеу қажет жағдайда $\bar{y}_j = \varphi_j(x_1, x_2, \dots, x_k)$, $j = \overline{1, m}$ тәуелділік функциясы коэффициентін анықтау үшін жеткілікті толық немесе толық емес факторлы тәжірибені таңдау жүзеге асырылады.

Стратегиялық жоспарды құру жайлы толығырақ ақпаратты [5, 12] алуға болады.

8.3.

КОМПЬЮТЕРЛІК ТӘЖІРИБЕНІ ТАКТИКАЛЫҚ ЖОСПАРЛАУ

Жүйенің машиналық модельімен тәжірибені тактикалық жоспарлау тәжірибе үшін бөлінген машина ресурстарын тиімді

пайдалану және стратегиялық жоспарлау кезінде құрылған тәжірибе жоспарымен белгіленген модель сынағын жүргізудің нақты әдістерін анықтау мәселелерімен байланысты. Сонымен қатар мәселелер шешіледі:

- бастапқы шарттың анықтау және олардың модельдеу кезінде орнатылған жетістікке әсерін анықтау;
- модельдеу нәтижелерінің дәлдігі мен нақтылығын қамтамасыз ету;
- жүйе модельдерімен имитациялық тәжірибенің автоматты түрде тоқтату ережелерін таңдау.

8.3.1. Орнатылған тәртіпке жетудің бастапқы шарттары мен олардың әсер етуі

Машиналық тәжірибені жүргізу кезіндегі бірінші мәселе модельді іске қосулық бастапқы шартының әсерінен машиналық модельдің бастапқы периодының бұрмалануына негізделеді. Бұл кезде N таңдама көлемінің өзгеруі кезіндегі бағаланатын параметр p (p параметрі ретінде қызмет көрсетуге өтінімді қабылдау ықтималдылығы алынған) қадімгі тәртібі 8.2 суретте көрсетілген.

Бастапқы периодтың алынатын деректің әсерін азайтудың үш негізгі жолы бар.

1. Ауыспалы периодтың деректер саны орнатылған тіртңп дерегінің санымен салыстырғанда мардымсыз болуы үшін, айтарлықтай ұзын есептегіш аралықты пайдалану, бірақ бұл кезде модельдеу уақыты біршама артады.

2. Модельдің орнатылған жұмыс тәртібіне жету үшін талап етілетін бастапқы аралық периодын қарастыруды алып тастау. Бұл жағдайда ауыспалы процес уақытын анықтау қиындық тудырады. Егер оны өте аз таңдаса, онда бастапқы шарттың әсері қалады. Егер оны тым үлкен таңдаса, онда пайдалы деректің бөлігі тасталынуының арқасында модельдеу уақыты артады.



Сур. 8.2. Параметр бағасының таңдама көлемімен өзгеру тәуелділігі

3. Қарапайымға жақын осындай орнатылған бастапқы шарттың күйін таңдау, және сол арқылы ауыспалы периодты азайту. Бұл жағдайда модельдеу кезінде жүйенің қарапайым күйін анықтау мәселесі туындайды.

Осылайша, осы әдістердің барлығы модельмен машиналық тәжірибені жүргізу кезінде ауыспалы процестің әсерін тек азайтуға, бірақ нөлге келтірмеуге мүмкіндік береді.

Одан әрі модельдеудің берілген дәлділігі мен нәтиженің нақтылығын қамтамасыз етуге мүмкіндік беретін таңдама көлемін анықтау мәселелерін қарастырамыз, ол [10] сипатталған.

8.3.2. Іріктеменің жиынтықтың орташа мағынасын бағалау

Есеп белгілі бір дәлдікпен және сенімділікпен минималды x уақыт ішінде орташа мәнінен кейбір зерттелген параметрдің белгісіз математикалық күту m бағалауын алудан тұрады. Орташа іріктеме, әртүрлі бастапқы жағдайлармен модельдеудің бірнеше имитациялық аралықпен алынған, кездейсоқ шама X ретінде қарастыруға болады (x іріктеуден іріктемеге өзгереді) және іріктеме мағынасының белгілері x_1, x_2, \dots, x_N — бірдей тәуелсіз таралған кездейсоқ шамалар сияқты X_1, X_2, \dots, X_N математикалық күтілімімен және орташа квадраттық ауытқуы σ . Алдымен біз тәуелсіздіктің болжамын және модельдік жауаптардың қалыпты таралуын қолданамыз. Бұл болжам ықтималдықтар теориясының орталық шекті теоремасын қолдануда негізделеді, оның кездейсоқ шамаларын таралуын мәні бірдей қабылдауы, бірдей таратулары ықтималдық бар көптеген тәуелді емес шамалардың үлкен сандар сомасы жиынтығы болып табылатын, қалыпты таратуға жақын мәндерінен тұрады. Көптеген жұмыстың аяғында тәуелсіздік пен бірдей тарату талаптары қажет емес екендігі көрсетіледі. Жиі бұл шағын салдардың көп сан сомасына жауап қайтармауы жеткілікті болады. Сондықтан кездейсоқ айнымалы мәндердің көп мөлшерін аддитивтік әрекетінің нәтижесі болып табылатын күрделі имитациялық модельдеуге жауап қайтармау айнымалысы шамамен қалыпты түрде

таратады. Оның нәтижесінде, егер кездейсоқ шама X жақсы таратылса, онда орташа іріктеме \bar{x} , тәуелсіз қадағалау бойынша табылған, сонымен қатар математикалық күтілімі жақсы таратылған

$$M(x) = \mu$$

және орташа квадраттық ауытқуы

$$\sigma(\bar{x}) = \sigma/\sqrt{N}.$$

Қатынастардың орындалуын талап етеміз

$$P(\mu - d \leq \bar{x} < \mu + d) = \beta, \quad (8.1)$$

Мұндағы x — іріктеме орташа мағынасы; d — бағаның дәлдігі; P — ықтималдылығы, $\mu \pm d$ аралығын құрайды (сенімділік, зерттеушінің айтуымен).

Берілген сенімділік β формуламен алынған x бағасының маңыздылығына байланысты

$$\beta = 1 - \alpha.$$

Егер математикалық күтілімі m белгілі бір дәлдікпен және сенімділік P (мәнінің деңгейі α) берілген, онда бұл дәлдікті қамтамасыз ететін минималды іріктеу көлемі формула бойынша анықталады

$$N = \frac{t_{\beta}^2 \sigma^2}{d^2},$$

мұндағы t_{β} саны теңсіздікпен анықталады $2\Phi(t_{\beta}) = \beta$ немесе $\Phi(t_{\beta}) = \beta/2$

Лапласа функциясының кестесі бойынша (бұл жерде

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{z^2}{2}} dz$$

Егер дәлділік d белгісіз болса және орташа квадраттық ауытқу σ , онда мәселені төмендегідей етіп орнатуға болады: іріктеу көлемі қандай болуы керек, ықтималдықпен қажетті бағалау үшін, $\beta = 1 - \alpha$ шектік табу $m \pm \Delta\sigma$, мұнда $\Delta\sigma$ — кейбір үлес σ (мысалы, $\Delta\sigma = \sigma/2$; $\sigma/4$; $\sigma/6$; $\sigma/10$; $\sigma/20$; ...).

Мысалы, кезінде $\Delta\sigma = \sigma/4$ және $P = 0,95$ ($\sigma = 0,05$) аламыз $\beta = 1,96$, онда

$$N = \frac{(1,96\sigma)^2}{(\sigma/4)^2} = 61.$$

Басқа жағдайларда тәжірибелік эксперимент арқылы шығыс

дисперсиясын анықтау және дисперсияның с2бағасын алу керек, содан кейін қажетті бақылаудың жалпы санын есептеңіз. Онда іріктеу көлемі келесі өрнекпен анықталады:

$$N = \frac{t_{\beta}^2 S^2(n)}{d^2},$$

мұндағы дисперсияны бағалау формула бойынша алынған

$$S^2(n) = \frac{1}{n} \sum_{i=1}^n x_i^2 - \left(\frac{1}{n} \sum_{i=1}^n x_i \right)^2,$$

Мұндағы n— сынамалы эксперименттегі іріктеу көлемі.

8.3.3. Чебышев теоремасын қолдану

Чебышев теңсіздігінде айтылады, бұл берілген белгілі бір санға k (кемінде бір) және ерікті іріктеу x_1, x_2, \dots, x_N көлемінде N кем дегенде $1 - 1/k^2$ өлшем үлесі жақын орналасады орташа мәнінен аспайтын қашықтықта - орташа квадраттық ауытқу. Бұл теңсіздік кез-келген таралуына жиынтығында әділ. Егер біз қалыпты таралу туралы болжамнан шықпасақ (әрқашан жеткілікті дәлдікпен қанағаттандырылмаса), Чебышев теңсіздігін қолдануға болады, ол келесі түрде болады

$$P(|x - \mu| > k\sigma) \leq \frac{1}{k^2}.$$

Шарт қоямыз, онда бағалау кезінде Харалықта болуы керек $\mu \pm \sigma/m$ ($m = 2, 4, 6, 10, \dots$) ықтималдылығымен $P = 1 - \alpha$:

$$P\left(|\bar{x} - \mu| > \frac{\sigma}{m}\right) \leq \alpha,$$

онда, теңдікті қолданып (8.1), аламыз

$$P\left(|\bar{x} - \mu| > \frac{\sqrt{N} \sigma}{m \sqrt{N}}\right) \leq \alpha = \frac{m^2}{N},$$

өйткені

$$k = \frac{\sqrt{N}}{m}; \frac{1}{k^2} = \frac{m^2}{N}.$$

Кесте 8.1. Әр түрлі мағынадағы қателік кезіндегі іріктеу көлемі

Ауытқу	Керекті іріктеу көлемі N	
	Орташа шекті теоремасы бойынша	Чебышев теоремасы бойынша
$\sigma/2$	15	80
$\sigma/4$	61	320
$\sigma/6$	138	720
$\sigma/8$	246	1 280
$\sigma/10$	384	2 000
$\sigma/12$	553	2 880
$\sigma/20$	1 537	8 000

Осы жерден керекті машиналық тәжірибенің іріктеу көлемін табамыз

$$N = \frac{m^2}{\alpha}$$

Алынған іріктеу көлемі қалыпты әлдеқайда көп, таралу жиынтық кезінде жеткілікті. Әр түрлі мағынадағы іріктеу көлемі кезінде $\alpha = 0,05$ ($\beta = 0,95$) кест. 8.1. келтірілген

8.3.4. Пайыздық қатынастарды бағалау

Көптеген жағдайларда A машина тәжірибенің мақсаты зерттелетін жүйенің жұмыс процесінің жай-күйімен анықталған оқиғаның $P = P(A)$ жағдайының ықтималдығын бағалауды алу болып табылады. Ықтималдылық санын бағалау ретіндержиілік $p = \Theta/N$, где Θ — тәжірибенің тиімді шығыс саны.

Жиілікті көрсетеміз $p = \Theta/N$ түрінде

$$\hat{p} = \frac{1}{N} \sum_{i=1}^N x_i,$$

мұндағы оқиғаның бақылау саны A берілгенді жүзеге асыру N -ненбақылау кездейсоқ шама болып табылады, мағынасын

қабылдайтын $x_i = 1$ ықтималдылықпен и $x_i = 0$ қосымша ықтималдылықпен $1 - p$.

Онда

$$\bar{X}(N) = p;$$

$$S^2(N) = \frac{p(1-p)}{N}.$$

Лаплас теоремасы бойынша, жеткілікті үлкен N шамасындағы $0/N$ жиілігін кездейсоқ шама ретінде қарастыруға болады, ол математикалық болжамы $X(N)$ және таңдаулы дисперсиясы $S^2(N)$ бар ықтималдықтарды бөлудің қалыпты заңымен сипатталады. Сондықтан ε бағалау дәлдігі және β нақтылығы N санымен байланысты болып келеді. ε дәлдігі және β нақтылығы бар бағалауды алуға қажетті таңдау көлемі осы арқылы анықталады.

$$P\left(p - \varepsilon < \frac{\theta}{N} < p + \varepsilon\right) = \Phi\left(\frac{p + \varepsilon - p}{\sqrt{p(1-p)}} \sqrt{N}\right) - \Phi\left(\frac{p - \varepsilon - p}{\sqrt{p(1-p)}} \sqrt{N}\right) = \beta,$$

P ықтималдығы туралы алдын-ала деректерді алу мүмкіндігі болмаған жағдайда абсолютті дәлдікті пайдалану өз мәнін жоғалтады. Мұндай жағдайларда модельдеу нәтижелерінің қатысты дәлдігін анықтау қажет болады:

$$N = \frac{t_{\beta}^2 p(1-p)}{\varepsilon^2}.$$

P мәні белгісіз болған жағдайда, тәжірибені тактикалық жоспарлау кезінде ерікті таңдалған n мәніне арналған алдын-ала модельдеу жүргізіледі. Алдын-ала модельдеу нәтижесі бойынша, $p_0 = \theta/n$, анықталады, одан кейін (8.2) қатынасы бойынша N өткізу саны есептеледі, ол үшін p орнына p_0 мәні қолданылады. Мұндай N бағалау процедурасы машиналық тәжірибе кезінде бірнеше рет орындалады.

$$\varepsilon_0 = \frac{\varepsilon}{p}.$$

Онда қатынастар (8.2) келесі түрде

$$N = \frac{t_{\beta}^2 (1-p)}{\varepsilon_0^2 p}.$$

Формула (8.3) сирек кездесетін оқиғалардың статистикалық

модельдеудің нақты ерекшелігін көрсетеді, бұл шағын р ықтималдықтарды жоғары дәлдікпен керекті бағалау үшін өте үлкен санды іске асыру қажет. Тәжірибелік жағдайда ықтималдылықты рет ретімен бағалау үшін 10-кмақсаттық санын ең болмаса $10k+1$ теңдей таңдап іске асыру.

8.3.5. Жиынтық дисперциясын бағалау

Жүйенің тиімділігінің көрсеткіші ретінде σ^2 дисперсиясын көрсетуге болады, бағасы s^2 ол машиналық тәжірибелердің жүргізу нәтижесі ретінде алынуы керек, кейбір сенімділік $\beta = 1 - \alpha$.

Талап етеміз, қатынастар орындалуын

$$P(s^2 - d \leq \sigma^2 \leq s^2 + d) = \beta.$$

Екі еселік теңсіздікке айналдырамыз

$$s^2 - d \leq \sigma^2 \leq s^2 + d$$

мәндес

$$s^2(1 - q) \leq \sigma^2 \leq s^2(1 + q).$$

Мұнда q — сан, бағалаудың жақындығының дәрежесін сипаттайтын s^2 шынайы дисперцияға σ^2 ; $q = d/s^2$.

Қолдануға тиімді болу үшін қарастырылған сатистиканы қолданамыз

$$\chi^2 = \frac{s^2(N-1)}{\sigma^2},$$

заң бойынша таралған χ^2 («хи-квадрат») $N - 1$ еркіндік дәрежесімен.

Таралу тығыздығы χ^2 келесі түрде болады

$$k_N(x) = \begin{cases} \frac{1}{2^{N/2} \Gamma\left(\frac{N}{2}\right)} x^{\frac{N}{2}-1} e^{-x/2}, & x > 0; N \geq 1; \\ 0, & x \leq 0. \end{cases}$$

Мұнда $\Gamma(\lambda)$ — гамма-функциясы, анықталған

$$\Gamma(\lambda) = \int_0^{\lambda} t^{\lambda-1} e^{-t} dt.$$

Атап айтқанда аргументтің жалпы мағынасы $\Gamma(n+1) = n!$

Үлкен N үшін орталық шектік теоремасына сәйкес қалыпты шамалардың таралуы $(\chi^2 - N)/\sqrt{2N}$, стандартты қалыпты таралуына ұмтылады

$$P\left(\frac{\chi^2 - N}{\sqrt{2N}} < x\right) \approx \Phi(x),$$

Осыдан теңдікті алуға болады

$$t_{\beta}^2 = \frac{q^2(N-1)}{2}$$

немесе

$$N = 1 + \frac{2t_{\beta}^2}{q^2},$$

мұнда t_{β} — стандартты қалыпты таралу квантильі, мәнін теңдеуден табамыз $2\Phi(t_{\beta}) = \beta$ Лапласа функциясының кетесі.

8.3.6. Автокорреляциялық деректер

Егер модельдің шығысындағы мәндері автокорреляцияланған болса, яғни кейінгі таңдалған мәндері алдыңғысына тәуелді болса, онда бұл іріктеуде тәуелсіз деректер іріктеуіне қарағанда аз ақпарат болады. Сонымен қатар талап етілетін іріктеу көлемі автокорреляция шамасына өте сезімтал.

Автокорреляция функциясын бағалау әдісін қарастырайық және оның параметрлік бағалауларын құрастыру кезінде қолдануға болады, олар корреляциялық деректердің жағдайда

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i;$$

$$\sigma_N^2 = \frac{\sigma_x^2}{N} \left[1 + 2 \sum_{p=1}^m \left(1 - \frac{p}{m+1} \right) \rho_{p,x} \right],$$

Мұнда σ_x^2 — жиынтық дисперсия; m — қарастырып жатқан автокорреляцияның максималды тереңдігі; $\rho_{p,x}$ - p -й автокорреляция

коэффициентті; $p = 1, 2, \dots, m$.

Корреляциялық коэффициенттерді бағалау үшін сынақ іріктеуімен тәжірибе жүргізу және формуланы қолдану қажет

$$\rho_{p,x} = \frac{\sum_{i=1}^{N-p} [(x_i - \bar{x})(x_{i+p} - \bar{x})]}{S^2}$$

Мұнда S^2 — жөнделген іріктеу дисперсиясы, негізінде алынған N сынау тәжірибелерінде өлшенген.

Минимальды қажетті іріктеу көлемін формула бойынша анықтауға болады

$$N = \frac{t^2 S^2 \left[1 + 2 \sum_{p=1}^m \left(1 - \frac{p}{m+1} \right) \rho_{p,x} \right]}{(\bar{x}d)^2}$$

где d — шынайы орташа мәннің салыстырмалы қателігі.

Есептелген корреляция коэффициенттерінің максималды тереңдігі m тәжірибеде таңдалған өлшемдердің N санының шамамен 10% құрайды.

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

- 1 Машиналық тәжірибені жоспарлау дегеніміз не?
- 2 Компьютерлік тәжірибені тактикалық жоспарлаудың мәні қандай?
- 3 Бастапқы жағдайлар модельдеу нәтижелеріне қалай әсер етеді??
- 4 Берілген дәлдік пен тәжірибенің минимальды құнын қамтамасыз ететін іріктеу көлемін анықтау мәселесі қалай шешіледі?
- 5 Іріктеу жиынтығының орташа мәні қандай дәлдікпен анықталады?
- 6 Оңтайлы іріктеу көлемін бағалау үшін Чебышевтің теңсіздігін қалай пайдалануға болады?
- 7 Белгілі дәлдікпен оқиғаның пайда болу ықтималдығын бағалау кезінде іріктеу көлемін қалай анықтауға болады?
- 8 Берілген дәлдікпен ауытқуды бағалау кезінде іріктеу көлемін қалай анықтауға болады?

9 Автокорреляциялық деректер үшін іріктеу көлемі қалай есептеледі?

9 Тарау

ИМИТАЦИЯЛЫҚ МОДЕЛЬДЕУ ПАКЕТТЕРІ

9.1.

ИМИТАЦИЯЛЫҚ МОДЕЛЬДЕУ ПАКЕТТЕРІНІҢ ТАҒАЙЫНДАЛУЫ

Имитациялық модельдерді құру кезінде оларды әзірлеу үшін қажетті кейбір типтік функционалды мүмкіндіктерді бөлуге болады [5].

Осындай мүмкіндіктерге:

- 0-ден 1 дейінгі интервалда біркелкі тараған кездейсоқ санды түрлендіру;
- берілген таралу ықтималдылығы бар кездейсоқ шаманы түрлендіру;
- модельді уақытты қозғалтуды ұйымдастыру;
- шығатын статистикалық деректерді жинақтау және нәтижелері бар есеп құру және с.с. жатады.

Модельдеуші бағдарламаларда жалпы функционалды мүмкіндіктердің болуы имитациялық модельдерді жасап шығару процесін біршама жеңілдету мен тездетуге мүмкіндік беретін имитациялық модельдеудің арнайы бағдарламалық пакеттерін немесе сжүйелерін әзірлеуге әкеп соқты.

Әмбебап бағдарламалау тілін қалданумен салыстырғанда имитациялық модельдеу пакетін қолдану бірқатар ерекшеліктерді береді [5].

1. Имитациялық модельдеу пакеттері имитациялық модельді құру үшін талап етілетін функционалды мүмкіндіктердің көпшілігін автоматты түрде ұсынады, ол бағдарламалау үшін қажетті уақытты және жобаның жалпы құнын біршама қысқартуға мүмкіндік береді.

2. Имитациялық модельдеу пакеттері имитациялық модельдерді құру үшін қалыпты ортаны қамтамасыз етеді. Олардың негізгі

модельдеуші құрылымы сәйкес бағдарламаның әмбебап тіліндегі құрылымына қарағанда имитациялық модельдеуге көбірек келеді.

3. Имитациялық моделі пакеттердің көмегімен құрылған модельдеу, әдетте, түрлендіруге және пайдалануға оңай.

4. Имитациялық модельдеу пакеттері неғұрлым жетілдірілген тетіктері табылған қателерді анықтау механизмдерін қамтамасыз етеді, өйткені олар көптеген түрлердің автоматты түрде қателерді тексеруін орындайды. Және де модельге құрылымдық компоненттердің үлкен санының қажет болмағандығынан, кез келген қатені жасау мүмкіндігі азаяды.

Сол уақытта әмбебап бағдарламалау тілдерін қолдану, сондай-ақ бірқатар артықшылықтарға ие [5].

1. Бағдарламалау тілдерін көптеген әзірлеушілер біледі, имитациялық модельдеу пакеттері туралы айту мүмкін емес.

2. Әмбебап программалау тілдерінде жазылған имитациялық модельдерінің жылдамдығы, әдетте жоғары, модельдеу пакеттерінің көмегімен жасалған модельдерге қарағанда.

3. Бағдарламалау кезінде әмбебап тілдері имитациялық модельдеу пакеттеріне қарағанда икемділікті қамтамасыз етеді.

4. Әмбебап бағдарламалау тілдерінің қолданылатын кезінде пайдаланылатын бағдарламалық жасақтама құны әдетте имитациялық модельдеу пакеттерінің құнынан төмен болады.

Осылайша, екі әдісті пайдалану артықшылығы бар, сондықтан модельді әзірлеуші олардың әрқайсысын мұқият қарастыруы керек.

Қазіргі күні бір бірімен әмбебаптылық деңгейімен, функционалды мүмкіндіктерімен, пайдалану саласы мен көптеген басқа да көрсеткіштер бойынша ерекшеленетін имитациялық модельдердің көптеген пакет саны әзірленген. Одан әрі басқа модельдеу пакеттерімен салыстырғанда бірқатар артықшылықтарға ие Pilgrim атты имитациялық модельдеу пакеттерінің бірін қарастырамыз.

9.2.

PILGRIM ИМИТАЦИЯЛЫҚ МОДЕЛЬДЕУ ПАКЕТІ

Pilgrim пакеті имитациялық объектілердің уақытша, кеңістіктік және қаржылық динамикасын модельдеуге арналған кең мүмкіндіктерді ұсынады [3]. Оның көмегімен дискреттік-үздіксіз модельдер жасауға болады. Мәтін үлгісінде стандартты C ++ тілін

қолданып кез келген блоктарды кірістіруге болады. Pilgrim жүйесіндегі модельдер құрастырылады және сондықтан жоғары жылдамдықпен әсер етуге ие, ол аса жылдамдатылған уақыт масштабында басқару шешімдерін жасау мен нұсқаның бейімделген таңдауы үшін маңызды. Компиляциядан кейін алынған нысан кодты жасалып жатқан бағдарламалық кешендерге енгізуге немесе тапсырыс берушіге беруге (сатуға) болады, өйткені модельдерді пайдалану кезінде Pilgrim пакетінің құралдық заты қолданылмайды.

Pilgrim имитациялық модельдеуінің пакетінің [3] сипатталған ұсынылымын келтірейік.

9.2.1. Модельдің негізгі нысандары

Модельденетін жүйе тұжырымдамасы негізделетін алты негізгі түсінік бар.

1. Кез-келген имитациялық модель сызбасын модельдің бағытталған бағаны ретінде ұсынуға болады, олардың биіктіктері қарапайым үрдістердің құраушылары болып табылады, ал доғалар модельденетін жүйедегі басқарушы әрекеттерді және өтініштер ағынының бағытын анықтайды. Қарапайым СМО сипаттайтын баған мысалы 9.1 суретте көрсетілген.

2. Транзакт – қандай-да бір қызмет көрсетуге формалды өтініш. Жаппай қызмет көрсету модельдерін талдау кезінде қарастырылатын өтініштерге қарағанда, транзакт динамикалық ерекше қасиеттер мен параметрлерден тұрады. Модель бағаны бойынша транзакттарды миграциялау жолдары бағана түйініндегі модель құраушыларының логикалық қызмет етуімен анықталады.

Транзакт келесідей әрекеттерді орындай алады:

- басқа транзактілер тобын тудыра алады;
- нақты түрдегі басқа транзактілерді жұта алады;
- ресурстарды қамтып, оларды бірнеше уақыт пайдалана алады, одан кейін босата алады;
- қызмет көрсету уақыттарын анықтай алады, өтілген жол туралы ақпаратты жинап, басқа транзактілердің ары-қарайғы жолдары туралы ақпараттарды жинай алады.

Транзактінің негізгі параметрлеріне жатады:



Сурет 9.1 Модель бағанының мысалы

- транзактінің бірегей сәйкестендіргіші;
- транзакт тиесілі болатын топ нөмірі (сәйкестендіргіші);
- транзакт қамти алатын және қандай-да бір уақытта пайдалана алатын әртүрлі ресурстар жиынтығы;
- транзакт өмірінің уақыты;
- басымдылық – теріс емес сан; басымдылық жоғары болған сайын, транзакт те басым болады (мысалы, кезекте тұрғанда);
- қандай-да бір қызмет көрсетуші құрылғыдағы қызмет көрсету параметрлері (ықтимал сипаттамаларды қоса алғанда).

Транзакт мысалдарына жатады: ақша аудару талабы; фирмада жұмыс орындауға өтініш; сатып алушы дүкенде.

3. Желі бағанының түйіндері транзактілер бойынша қызмет көрсету орталығынан тұрады (жаппай қызмет көрсетуде міндетті емес). Транзактілер түйіндерде кідіруі, қызмет етілуі, жаңа транзактілер топтарын тудыруы, басқа транзактілерді жоюы мүмкін. Әрбір түйінде есептеу үрдісі тарапынан тәуелсіз үрдіс туындайды. Есептеу үрдістері параллельді орындалады және бір-бірлерін байланыстырады. Олар бірегей модельді уақытта, бір кеңістікте орындалады, уақыт, кеңістікті және қаржылық динамиканы есепке алады. Түйін мысалдары: өндірістік аумақ (жөндеу); билет сатуға арналған касса; ресурстарды бір кеңістіктен екіншісіне ауыстыратын тасымалдау құралы; ресурстар қоймасы.

4. Оқиға ретінде бір транзактінің түйінінен шығу қарастырылады. Оқиға әрдайым уақыттың белгілі кезеңдерінде пайда болады. Олар кеңістік нүктесімен де байланысуы мүмкін. Модельдегі көршілес оқиғалар арасындағы аралық – бұл әдетте кездейсоқ шамалар. Модельді дайындаушы оқиғаларды қолымен басқара алмайды (мысалы, бағдарламадан). Сондықтан, оқиғаны басқару функциясы арнайы басқару бағдарламасына беріледі – модель құрамына автоматты түрде кіретін үйлестірушіге.

5. Ресурс – модельдеу үрдісіндегі табиғатына тәуелсіз үш ортақ параметрмен сипатталуы мүмкін: қуаттылықпен, қалдықпен және тапшылықпен. Ресурс қуаттылығы – ресурсты бірліктердің максималды саны, оларды әртүрлі мақсатта пайдалануға болады. Ресурс қалдығы – транзактілерді қанағаттандыру үшін пайдалануға болатын бос бірліктер саны. Ресурс тапшылығы – аталған ресурс бойынша кезекте тұрған транзактілердің жиынтық өтінішіндегі ресурс бірлігінің саны.

Әдетте ресурстардың үш негізгі түрін бөліп қарастыруға болады: материалды, ақпараттық және қаржылық.

6. *Кеңістік* - декарттық, географиялық жазықтық (басқаларын да енгізуге болады). Түйіндер, транзактілер және ресурстар кеңістік нүктелеріне байланысып, сонда тасымалдануы мүмкін. Модельді ішкі жүзеге асыру экономикалық үрдістерді ұсынудың бағытталған-объективті тәсілін пайдаланады. Транзактілер, түйіндер, жағдайлар мен ресурстар – имитациялық модельдің негізгі объектілері. Осы объектілердің өзара әрекеттері баған құрылымымен анықталады.

Түрлі модельдеуші жүйелерде баған түйіндерін ұсынудың әртүрлі тәсілдері болады. Бұл осындай жүйелердің ерекшеленетін қасиеттерімен байланысты болып келеді. Мысалы, GPSS жүйесіндегі түйіндер блоктар деп аталады; блоктардың әртүрлі блоктарының саны жүзден асады, бұл модель бағанын қабылдауды күрделендіреді. Pilgrim пакетінде түйіндердің 17 түрі болады, олар GPSS барлық блоктарын қамтиды және қосымша құралдарды ұсынады, олар GPSS жүйесінде болмайды:

- үздіксіз үрдістермен жұмыс істеу мүмкіндігі;
- кеңістікті динамиканы модельдеу;
- ақша және материалды құндылықтардан тұратын ресурстармен жұмыс істеу, бухгалтерлік шот есептері, банктік шоттар.

Модель бағанын «оқуға» мүмкіндік беретін түйіндер мәнінің жүйесі болады. Әрбір түйін графикалық мәннен, функционалды атаудан, ерікті бірегей нөмірден және ерікті атаудан тұрады (мысалы, атауы – serv, нөмірі -123, аты –«Мастерская»). Транзактілер жолдары доғалармен анықталады - бағыттаушысы бар тұтас сызықтардан.

9.2.2. Имитациялық модель түйіндерінің түрлері

Имитациялық модель түйіндері модельденетін бағдарламаны жазу кезінде пайдаланушымен белгіленетін параметрлермен сипатталады. Түйін параметрлері тұрақты және ауыспалы мәннен тұрады. Кез-келген түйіннің бірінші параметріне p1 модельдегі символдық атауы, түйіннің логикалық бағыты жатады. Ары қарай Pilgrim пакетінің кейбір түйіндері және оның параметрлерінің бағыттары сипатталған.

ag түйіні – транзактілер генераторы. ag түйіні шексіз сыйымдылықтан тұратын транзактілер генераторынан құралады. Ол имитациялық модель транзактілерін құру үшін қолданылады. Транзактілер белгілі-бір уақытта пайдаланушымен бір-бірлеп түрленеді. Генератор тапсырмасы үшін ag (p1,p2,p3,p4,p5,p6,p7,p8). Параметрлер келесідей мәннен тұрады:

p1 – түйіннің символдық атауы (баған ұзындығы 14 символға дейін);

p2 –генератор-түйіннің нөмірі (int);

p3-әрбір түрленетін транзактіге арналған басымдылық, диапазондағы сан -1.....32767 (сан үлкен болған сайын, басымдылық та жоғары болады), егер басымдылық керек болмаса, p3=none.

p4-екі тізбекті түрленген транзактілер арасындағы уақыт аралығын бөлу функциясының түрі болып табылады. Шартты мәндер пайдаланылады: norm-қалыпты бөлу; unif- бірдей бөлу; expo-экспоненциалды бөлу; erln – Эрланг бойынша бөлу; beta-үшбұрышты бөлу; none-анықталған шама болып табылатын аралық.

p5-екі тізбекті түрленген транзактілер арасындағы уақыт аралығының математикалық болжамы (float), p4 =norm, expo, unif. Осы интервал қосылғышының математикалық болжамы (p4 = erln), немесе интервалдың минималды мәні (p4 = beta) немесе осы интервалдың тұрақты шамасы (p4 = none).

p6- бөлу функциясының түріне тәуелді болатын шама (float): орташаквадратты ауытқу (p4 =norm), немесе орташа шамада есептелген максималды ауытқу (p4 = unif), немесе zero (p4 = expo,none), немесе қосылғыш интервалдар саны (p4 = erln, бұл жағдайда p4> 0), немесе екі тізбекті түрленген транзактілер арасындағы уақыт интервалының болжамды мәні (p4 = beta);

p8-түрленетін транзакт берілетін түйін нөмірі.

Генератор параметрлері cheg командасының (пәрменінің) көмегімен модельмен жұмыс істеу кезінде өзгеруі мүмкін.

Serv түйіні – абсолютті басымдылықтары бар көпканалды қызмет көрсетуші аспап. Serv түйіні – модельді уақыт ішінде транзактілердің қандай-да бір қызметін көрсететін қызмет көрсетуші аспап.

Имитациялық модель тарапынан «қызмет көрсету» уақыттың белгіленген аралығында транзактінің кідірісімен түсіндіріледі. Сервер – көпканалды қызмет көрсетуші аспап, ол абсолютті басымдылықтар ережелері бойынша жұмыс істейді немесе оларсыз

істейді, «үзілген» транзактілер үшін стегтен тұрады (қатысты басымдылықтар ержесі queue түріндегі түйінде орындалады – кезек).

Имитациялық модель тарапынан «қызмет көрсету» уақыттың белгіленген аралығында транзактінің кідірісімен түсіндіріледі. Сервер – көпканалды қызмет көрсетуші аспап, ол абсолютті басымдылықтар ережелері бойынша жұмыс істейді немесе оларсыз істейді, «үзілген» транзактілер үшін стегтен тұрады (қатысты басымдылықтар ержесі queue түріндегі түйінде орындалады – кезек).

Сервер (p1,p2,p3,p4,p5,p6,p7,p8) түйінді функциясының көмегімен анықталады. Параметрлер келесідей мәнге ие болады:

p2 – қызмет көрсетуші каналдар саны (int), $1 < p2 < 32767$;

p3-қызмет көрсету тәртібі: abs - басымды транзактіден аз үзінді қызмет көрсетуден тұратын басымдылықты, немесе none – басымдылықсыз. p1,p4..... p8 параметрлерінің бағыты ag түйіні параметрлерінің бағытымен сәйкес келеді. Егер p3= abs белгілейтін болсақ, онда басымдылықсыз транзактілермен үздіксіз жұмыс істеудің екі мүмкіндігі бар: оларға толығырақ қызмет көрсетіледі немесе ұзу сипаты қызмет көрсетуді жанартуды талап етеді. Мысалы, егер компьютерде мәтін теру жүзеге асырылса, онда қорек көзінің аяқ асты өшуі (басымдылықты транзакт ретінде түрленеді) мәтінді теруді басынан бастауға итермелейді (басымдылықсыз транзакт). Транзактінің «қызмет көрсетуі» әдепкі қалпы бойынша қосымша қызмет көрсету режимінде жалғасатын болады.

Дегенмен, serv функциясының алдында иелену операторы орындалатын болса, t-ga=again, онда транзакт қызмет көрсетуді қайтадан бастау белгісін алады. Serv шыққаннан кейін, бұл белгі жоғалады.

Serv түйініндегі транзактілердің қызмет көрсетуінің орташа уақыты және ондағы кідірістердің орташа уақыты – әртүрлі уақыттар. Транзактінің қызмет көрсетуіндегі кідірістер – қызмет көрсетудің таза уақыты, ал түйінде басымдылықсыз транзактінің кідіріс уақытында ол қызмет көрсету уақыты және осы транзактінің үзілетін күйдегі уақыты ретінде енеді (яғни, басымдылықты транзактінің қызмет көрсетуі кезінде).

queue түйіні – қатысты басымдылықтары бар кезек. queue түйіні транзактілер кезегін модельдейді. Бұл кезек екі ереженің біреуі бойынша құрылады: немесе транзактілер түсу тәртібі бойынша реттеледі, немесе, одан басқа, басымдылықты транзактілер кезектің басына жақын орналасады, ал басымдылығы төмен транзактілер –

соңына қарай орналасады. Екінші жағдайда қатысты басымдылықтар ережесі жұмыс істейді. Аталған түйінді сипаттау үшін queue функциясы пайдаланылады (p1,p 2,p 3). p1 және p 3 параметрлері түйіннің символдың атауларын және бағандағы келесі түйін нөмірін анықтайды, ал p 2 кезекті ұйымдастыру түрін белгілейді. p 6 кезекті ұйымдастыру түрін белгілейді (егер p2=prty болса, онда басымдылықты кезек қолданылады, ал егер p2=none болса, онда басымдылықсыз қолданылады).

Term түйіні – транзактілер терминаторы. Term түйіні-транзактілер терминаторы, оның бағыты транзакті кіретін модельден шығарады және оның өмір сүру уақытын бекітеді, осы транзактінің генератордан шығуынан бастайды. Аталған түйінді сипаттау үшін term функциясы (p1) пайдаланылады, p1 түйіннің символдық атауын анықтайды.

Key түйіні-транзакт жолындағы клапан. Key түйіні – «шлагбаум» қағидасы бойынша жұмыс істейтін клапан немесе кілт. Клапан жабық болғанда, транзакт оған кіре алмайды. Егер клапан ашық болса, онда транзакт кідіріссіз келесі түйінге енеді. Мұндай түйіннің жабық күйде болуының орташа уақыты автоматты түрде есептеледі. Мұндай клапанды немесе кілтті басқару hold және rels пермендері арқылы орындалады.

Аталған түйін key функциясының көмегімен (p1,p2) сипатталады, оның параметрлері түйіннің символдық атауын және бағандағы түйін нөмірін анықтайды.

Creat түйіні – транзактілердің басқарылатын генераторы. Creat түйіні транзактілердің жаңа тобын құру үшін қолданылады. Барлық транзактілер қандай-да бір топқа жатады. Қарапайым генератордан шығатын транзактілер (ag) 0 нөміріндегі топқа жатады. Басқарылатын генераторды сипаттау үшін түйінді функциялар қолданылады: creat (p1,p2,p3,p4,p5,p6). Аталған түйіннің қызмет ету логикасы келесідей:

■ түйін арқылы туындатушы транзакт өтеді, ол f1 тобына жатады және келесі p6 түйініне түседі;

■ бір уақытта түйінде p3 жаңа транзактілердің түрлері f1= p2 нөмірімен түрленеді, олар p5 түйініне бағытталатын болады. f1 және f2 топтарының нөмірлері сәйкес келуі мүмкін.

Қалған параметрлердің бағыттарын келтірейік. p1 параметрі түйіннің символдық атауын анықтайды. p4 параметрі келесідей мәндерден тұрады: тудырушы транзактінің параметрлерін тираждау

үшін әрбір тудырушыға, немесе none – әрбір тудырушы транзактіге нөлдік мәндер параметрлерін беру үшін қолданылатын мән.

Delet түйіні – транзактілердің басқарылатын терминаторы. Delet түйіні топтардың кейбір диапазондарына жататын транзактілер тобын жою үшін қолданылады. Аталған түйін delete функциясымен сипатталады (p1,p2,p3,p4,p5,p6).

Бұлт түйіндердің қызмет көрсету логикасы келесідей: түйінге p4 тобының жоюшы транзакті кіреді және сонда p5 түскенге дейін болады: $p2 < \text{Number} < p3$, оларды бірдей жоя алады. Аталған түйінде жұтылатын транзактілердің уақыты бекітіледі. Транзактілердің талап етілетін саны түскеннен кейін транзакт p6 түйініне ауысады.

Send және direct түйіндері – бухгалтерлік шоттары бар операция. Pilgrim пакетінің негізгі объектілері (түйіндер, транзактілер, жағдайлар) кәсіпорынның (фирманың) қаржылық динамикасын сипаттау үшін ыңғайлы болып келеді. Бухгалтерлік есеп шотын (қосалқы шот) сипаттау үшін send түйіні қолданылады; түйін нөмірі і деп санайық. і түйініне кіретін транзакт - і шотынан басқа шотқа, белгілі бір сомада жазба жүргізуге өтініш. Жазбаны жүзеге асыру үшін і шотында (яғни і түйінінде) талап етілетін сома болу керек. Мұндай сома болмаған жағдайда, транзакт і шотына жеткілікті қаржының түсуін күту режиміне өтеді. Басқа сөзбен айтқанда, і нөмірі бар түйін бухгалтерлік жазбаны қалыптастырады, бұл транзактілердің арнайы кезегі болып табылады.

Келесі send түйіні міндетті түрде direct түйінінен тұруы керек, ол «қаржылық директор» рөлін атқарады. Direct түйіні клапан болып табылады, сол арқылы жазба орындауға болатын транзактілер ғана өте алады. Direct бір түйіні модельдегі send барлық түйіндеріне қызмет көрсете алады.

Send түйінін сипаттау үшін send функциясын пайдалану қажет (p1,p2,p3,p4,p5), олардың параметрлері келесі мәндерден тұрады: p1- түйіннің символдық атауы; p2-аталған сома аударылатын түйін-шот; p3-аталған сома өлшемі (double түрі). Қаржылық құралдарды өлшеу бірліктері – кез-келген валютада болады (рубльдер, долларлар және т.б). Нүктеден кейін міндетті түрде бір немесе екі санды белгілеу қажет - өлшемдердің пайдаланылатын бірліктерінің үлестері. Мысалы: 1 000 000.00 (бір млн руб.00 тиын):

p4- басымдылықтармен жұмыс істеу мүмкіндігін анықтаушы параметр (prty мәні) немесе осы жазбалардағы қажеттіліктердің пайда болу кезеңдері бойынша жазбаларды қатаң реттеу (none мәні).

Егер рrty мәні белгіленсе, жазба сомасы аз болған сайын, сұраныс та басым болады;

p5-direct түріндегі түйін нөмірі, ол қаржылық менеджментті жүзеге асырады және қажет болған жағдайда жазбалар жүргізеді.

Send әрбір түйінінде saldo ішкі атрибуты болады, ол і шотындағы қаржы қалдықтарын көрсетеді. Бухгалтерлік есеп шотындағы қаржының жетіспеушілігі defic атрибутында болады. Егер і түйініндегі saldo атрибуты нөлдік мәннен тұрса, бұл жазба тудыратын түйінде транзактілер болады (бір немесе бірнеше), осы транзактілермен талап етілетін соманың жиынтық тапшылығы defic атрибутында автоматты түрде көрінеді. Send түйінін басқару үшін assign пәрмені қолданылады. Оның көмегімен бухгалтерлік шоттағы қаржы қалдығын өлшеуге болады. Send түріндегі кезектерге қызмет көрсету бір немесе бірнеше «қаржылық директор» түйіндерін пайдалану арқылы орындалады. Мұндай түйінді сипаттау үшін direct (p1,p2) функциясы қолданылады. Бұл жердегі p1 параметрі түйіннің символдық атауын анықтайды, ал p2 параметрі транзактінің қабылдаушы-түйінінің нөмірін анықтайды, ол бухгалтерлік жазбаны орындайды.

Attach және manage түйіндері – ресурс қоймаларымен жұмыс істеу. Pilgrim пакетінде имитациялық модельдерді құру кезінде модельденетін үрдістердің өту уақытын ғана емес, сонымен бірге, осы үрдістермен пайдаланылатын, жұмысқа қажетті ресурстарды қолдану да қарастырылады. Ресурс ретінше қоршаған әлемнің кез-келген объектісі есептеледі (шикізат, құрылғы, персонал). Модельдегі ресурс көлемі бүтін сандарда өлшенеді. Үрдіспен пайдаланылатын ресурстардың барлық түрлері арнайы attach түйін-қоймаларында сақталады. Attach түйінін транзактімен қараған кезде ондағы сақталатын ресурстың бірлік саны алынады. Егер қоймадағы ресурс саны аз болса (транзакті талабынан), онда ресурс көлемі көп болмағанға дейін ол түйінде қала береді, яғни, attach түйіні ресурс алу бойынша өтініштер кезегінде қалады. Өтініштер кезегі қарапайым болуы мүмкін, және өтініш басымдылығын есепке алуы да мүмкін. Басымдылық жоғары болған сайын, талап етілетін ресурс көлемі де аз болады. Модельдегі attach түйінінен кейін міндетті түрде manage түйіні тұруы керек, ол қойманы басқарады. Бұл түйін әрбір транзакт үшін attach түйініне келетін ресурстың талап етілетін санын тексереді. Егер ресурс жеткілікті болса, онда транзакт manage түйіні арқылы ары қарай өтеді. Әйтпесе транзакт attach түйінінде қалады.

Attach түйіні attach функциясымен орындалады (p1,p2,p3,p4). Оның параметрлерінің мәні келесідей:

p1-түйіннің символдық атауы;

p2-ресурс бірлігінің талап етілетін саны;

p3-pty немесе none – өтініштер басымдылығы есебінің жалаушасы. Send түйіні үшін де, өтініштер көлемі аз болған сайын, оның басымдылығы да жоғары болады;

p4-manage түріндегі түйін нөмірі, аталған қойманы басқарады.

Ресурс қалдығы және тапшылық rsal және rdef атрибуттарында бекітіледі. Тапшылық барлық кезекте тұрған транзактілердің attach түйініндегі өтініштердің жиынтық көлеміне тең болады (ресурстың жетіспеушілігіне байланысты қызмет көрсетілмейтіндер).

Транзактілерге қызмет көрсетумен manage түріндегі түйін айналысады – ол қойманы басқарады. Аталған түйінді сипаттаушы функция manage түрінен тұрады (p1,p2). Аталған функцияның p1 параметрі түйіннің символдық атауын анықтайды, ал p2 параметрі транзакт қабылдаушысының нөмірін анықтайды.

Қоймада алынған транзакт ресурс модель бойынша «тасымалданады» және оны қоймаға қайтара алады. Транзакт ресурсты detach пәрменінің көмегімен модельдің кез-келген түйінінен қайтаруға болады. Ресурсты қайтару міндетті емес. Қоймадағы ресурс көлемін өзгерту үшін supply пәрмені қолданылады.

9.2.3. Тораптарды басқару пәрмендері

Тораптарды басқару командалары қажет болған жағдайда кез келген түйінде пайдаланылуы мүмкін. Әрбір үлгі торабы кез келген басқа түйінді және өзі басқара алады. Команда әрбір транзактті көрсеткен түйінді енгізген сайын іске қосылады.

Блокталған delet торапты - freed пәрменімен босатыңыз.Freed (p1) командасы delet түйінінің блокталуымен күресуге арналған. Ол келесі логика бойынша жұмыс істейді.

Мысалы, p1 санымен delet торабында транзакт «тоқталды» (яғни, жойылатын транзакциялардың қажетті саны ұзақ уақытқа қабылданбайды) деп есептейік.Freed функциясы арқылы өтетін ағымдағы транзакт дереу қосалқы транзакті жасайды және оны p1 торабына жібереді.Бұл қосалқы транзакт кептеліп, «сыртқа шығады», deletқалыпты жағдайға алып келеді, ал өзі өледі. Қосалқы мәмілеге

келу фактісі delet-ге тіркеледі және статистикалық нәтижелерде көрінеді, бірақ оның өмір сүру уақыты нөлге тең. Кептелген транзакция келесі delet түйініне жіберіледі.

Ag транзакт генераторын көшіру - chег пәрмені. Генераторды P2 нөмірімен жаңа параметр мәндеріне орнату үшін chег(p2, p3, P4, P5) командасы қолданылады. Модельдік уақыт мағынасында осындай қайта жүктеу «бірден» болады. Бұл пәрменнің барлық параметрлері генератордың ag параметрлері (генератордың символдық атауы командада көрсетілмеген) сияқты бірдей мақсатқа ие.

Клапанды басқару key - hold және gels командалары. Hold (p1) және gels (p1) командалары кез келген басқа түйіндерден p1 санымен клапанды басқару үшін қолданылады. Holdорындаудан кейін, клапан p1 күйі «жабық», ал кейін gels - «ашық» жағдайда алады.

Бухгалтерлік шотты басқару - команда assign. Assign (p1, p2, p3) командасы p1 (double) ақша p1 сомасын орналастырады. Егер p2 операциялық кодының мәні add мәніне тең болса, онда шоттың балансы p3 (қосу режимі) сомасына көбейтіледі; егер p2 = none болса, онда шоттың балансы тең p3 (ауыстыру режимі) болады.

Ресурс қоймасын басқару - supply және detach пәрмендері. Supply (p1, p2, p3) командасы p1 қоймасындағы ресурстық бірліктердің p3 (long) мәнін орнатады. Егер p2 операциялық кодының мәні add-ге тең болса, қоймадағы ресурстың қоры p3 (қосу режимі) сомасына көбейтіледі; егер p2 = none болса, онда ресурстың қоры p3 (ауыстыру режимі) тең болады.

Detach (p1, p2) командасы p1 нөміріне p1 нөмірімен қоймадағы ресурс бірліктерінің санын қайтарады. Моделді дұрыс пайдалану үшін ресурстың көрсетілген көлемі бұрын қойма пәрменінде көрсетілген қайтару мәмілесі бойынша қабылданады (транзакция қайтарылады, ол detach пәрменін іске қосады).

9.2.4. Транзакт параметрлері және тораптардың жай-күй параметрлері

Транзакт параметрлері бар, олардың кейбіреуі пайдаланушыға қол жетімді. Транзакт параметріне кіру тек модель түйініне енген сәтте мүмкін болады және торапқа көшірудің атауы болып табылатын жүйе айнымалы t арқылы жасалады.

Мәміленің негізгі параметрлері болып табылады: $t \rightarrow iu0$, $t \rightarrow iu1$, $t \rightarrow iu2$, $t \rightarrow iu3$ - пайдаланушының еркін бүтін параметрлері (мәміленің жай-күйі туралы ақпаратты сақтау үшін пайдаланылады);

$t \rightarrow ru0$, $t \rightarrow ru1$, $t \rightarrow ru2$, $t \rightarrow ru3$ - өзгермелі нүктесі бар ауыспалы

нысаны бар ерікті пайдаланушы параметрлері;

$t \rightarrow ga$ - қайтадан мәміле жасаудың белгісі, ол `serv` функциясын қараған кезде көрсеткендей, меншіктеу оған ғаламдық тұрақты операциясымен `again` тағайындалуымен анықталады;

$t \rightarrow pr$ - мәміленің басымдығы;

$t \rightarrow ft$ - бұл транзакт тиесілі отбасының саны.

Модельдің күтпеген мінез-құлқынан қорықпай, тек қана өзгеріске ғана ие бола алмайды, тек қана, $t \rightarrow iu1$, $t \rightarrow iu2$, $t \rightarrow iu3$, $t \rightarrow ru0$, $t \rightarrow ru1$, $t \rightarrow ru2$, $t \rightarrow ru3$, $t \rightarrow ga$, және , кейде $t \rightarrow pr$ және $t \rightarrow ft$. Басқа параметрлерін өзгерту тек `Pilgrim`-ге ғана сеніп тапсырылуы тиіс.

`Pilgrim` пакетіндегі үлгі түйіндері, сондай-ақ параметрлерге ие. Кейбір түйін параметрлері талдау үшін пайдаланушыға қол жетімді (бірақ оларды өзгерту үшін емес). Параметрлер жүйелік `addr` алқабында бар. Параметрге

`addr <n> -><параметр>`

мекен-жайының өрнегі арқылы кіруге болады, мұндағы `<n>` - үлгідегі түйін нөмірі.

Кесте 9.1. Тораптар параметрлері

Аты	Параметр мақсаты
<code>nc</code>	Тораптағы арналардың саны
<code>na</code>	Модельдік уақытында түйін арқылы өткізілген транзакциялар саны
<code>tn</code>	Модельдік уақытында торапта болатын мәмілілер саны
<code>ts</code>	Қазіргі уақытта есептелген орташа жұмыс уақыты
<code>°p</code>	Кей типіндегі тораптың күйі. Бұл клапан ашық болғанда <code>true</code> мәнін қабылдайтын бүтін айнымалылар, немесе егер ол жабық болса, <code>false</code>
<code>se</code>	<code>Delete</code> типті түйіннің күйінің ерекшелігі. Бұл торапта деструктивті операция болмаған кезде <code>nil</code> мәнін қабылдайтын бүтін айнымалылар
<code>saldo,</code>	<code>Saldo</code> үлгідегі торабына байланысты есеп
<code>defic</code>	шотындағы қаражат тапшылығы және қалдығы
<code>rsal,</code>	Қоймадағы ресурстық тапшылық және қалдық
<code>rdef</code>	<code>rdef</code> торабы үлгідегі <code>attach</code> -қабайланысты

Түйіндер параметрлерінің аттары мен тапсырмалары Кестеде келтірілген. 9.1 Кез келген түйіннің параметрлерін талдау үшін оның <n> нөмірін пайдалануға болады. Мысалы, егер транзакт delet типті түйіннің жағдайын 5 санымен талдауға тура келсе және жоюшы транзакция болмаса, операторды орындаңыз, ол келесі өрнектен өтуі керек:

```
if (addr [5] -> se == nil) < оператор>;
```

Бұл жағдайда транзакттың өзінде бұл кезде еркін түйінде болуы мүмкін.

9.2.5. Кестелік модельдеу нәтижелері

Модельдеу нәтижелері туралы есеп бір немесе бірнеше беттен тұратын кесте ретінде көрсетіледі. Кесте модельдік бағдарламада модельдің авторы көрсететін жеке файлда жазылады. Осы кестенің өрістерін жіктеу 9.2. кестеде келтірілген.

9.2-кесте. Модельдеу нәтижелерімен кесте өрістерін тағайындау	
Баған	Толық Жазылуы
№	Торап нөмірі
Торап түрі	Торап түрі (ag, serv, key, queue, term, creat, delet, proc, dyna, send, direct, attach, manage)
Нүкте	Модельдеу аяқталған кезде creat, delet немесе процестің торабы орналасқан кеңістіктегі соңғы нүктенің саны
Жүктеу, % жол, км	Serv, key немесе proc типті түйіндерді пайыздық көрсеткіште пайдалану коэффициенті есептеледі. Key түйіндері транзакциялар арқылы өтетін кезде пайдаланылады. Attach торабы үшін бұл ресурсты пайдалану коэффициенті
M [t] орташа	Түйіннің түріне байланысты түйінде немесе басқа уақыт аралығындағы транзакцияның кешігу уақытының орташа мәні: l) d - түйінге жұмсалған орташа уақыт (абсолютті басымдықты ережені пайдалану кезінде артық емес мәмілелерге қызмет көрсету уақытынан ұзағырақ болуы мүмкін);

9-2. кесте жалғасы

Баған	Толық Жазылуы
М [t] орташа	2) queue, send және attach - кезектегі орташа кідіріс уақыты; 3) ag - екі генерацияланған транзакт арасындағы орташа уақыты; 4) term немесе delet - жойылған мәмілелердің орташа мерзімі; 5) key - жабық жағдайда орташа уақытты өткізу уақыты
C2 [t] вариациясы	Кешігу уақытының дисперсиясының кешігу уақытының орташа мәнінің квадратына қатынасы (вариация коэффициенті, квадратқа бой көтергені)
Кіріс есептегіші	Торап арқылы жасалған транзактылардың саны немесе жасалынған транзактылардың саны (ag немесе creat үшін) немесе жойылған (term немесе delet үшін). Key үшін берілген hold пәрмендерінің саны
Арналардың саны	Тораптағы арналардың саны
Транз – актілердің қалдығы	Модельдеу аяқталған кезде түйінде сақталған транзактылардың саны
Осы сәтте торап жағдайы	Модельді іске қосу соңында түйіннің жағдайы: келесі транзактіні енгізу үшін түйін «ашық» немесе «жабық» болуы мүмкін (тораптар serv және key).Сервер ашық, онда кемінде бір бос арна бар. Send және attach түйіндері үшін кестенің қалған бөлігі және қор тапшылығы (тиісінше «S» және «D» әріптері)

Сонымен осы стандартты Pilgrim құралдарымен қатар пайдаланушы әрқашан өздерінің құжаттарын басып шығаруға немесе олардың нәтижелерін модельдеу үрдісінде де, соңында да шығаруға мүмкіндік алады. Бұл C ++ тілінде жазылған және Pilgrim бағдарламасының сәйкес нүктелерінде орналасқан пайдаланушы анықтайтын операторлар немесе функциялар көмегімен жасалады.

9.2.6. Модельдерді сипаттау тілі

Pilgrim бағдарламасының жалпы құрылымы. Pilgrim пакетіндегі имитациялық моделін сипаттайтын бағдарлама құрылымы келесідей:

```
#include <pilgrim.h>
[констант пайдаланушының сипаттамасы]
[ауыспалы пайдаланушының сипаттамасы] forward {
    modbeg (...);
    ag (...);
    [бастапқы пәрмендер assign және supply]
    [бастапқы пайдаланушының функциясы]
    network (... )
    {
        < блок торабының сипаттауы >
        fault (...);
    }
    [Пайдаланушының аяқтау функциялары]
    modend (...);
    return 0;
}
```

Модельдің бірінші жолы - «Pilgrim» жүйелік файлын қосу командалары. Содан кейін пайдаланушы C++ тілінің ережелеріне сәйкес қажетті тұрақты және айнымалыларды сипаттай алады.

Modbeg функциясынан кейін барлық ag генераторларын сипаттау жүреді. Үлгінің басқа түйіндері түйіннің сипаттамасы блоктар көмегімен сипатталады. Нөмірлеуге қарамастан кез келген тәртіпте түйіндерді сипаттауға болады.

Үлгіні инициализациялау modbeg функциясы. Модельдеу бағдарламаларын іске қосу операторы modbeg (p₁, p₂, p₃, p₄, p₅, p₆, p₇, p₈, p₉). Бұл функцияның дәлелдері келесі мағынаға ие:

p₁ - бұл модельдің символикалық атауы;

p₂ - бұл үлгі түйіннің максималды саны (int), $2 \leq P_2 \leq 1,024$;

P₃ - модельдеуді орындау қажет уақыт кезеңі (float);

p₄ - жалған кездейсоқ шамаларды (long) сенсорларын теңшеу үшін пайдаланылатын ерікті бүтін сан. P₄ ретінде компьютердің таймерінің мәнін пайдалануға болады, оның қол жетімділігі келесі пішінге ие: (long time(NULL)). Бұл жағдайда түрлі уақыттарда

жүргізілген модельдеу нәтижелері әр түрлі болады;

p_5 - кеңістіктік модельдеу режимінің белгісі: earth - Жер беті, plane - декарттық жазықтық, cosmos - еркін кеңістік. Егер кеңістіктік модельдеу пайдаланылмаса, none пайдалануға болады;

p_6 - кезектердің біреуінің (queue түрінің торабының) саны, ол осы кезектегі кідірістердің динамикасын нәтижелерді графикалық көрсету арқылы талдау үшін уақытында қадағалауға тиіс. Мысалы, 3 санын көрсетсеңіз және модельде бұл нөмірмен кезек бар болса, онда осы кезектің динамикалық кескінін тікелей имитациялауға болады;

p_7 - бұл кеңістікте де, нәтижелерді графикалық көрсету кезінде де бақылануға тиіс процестердің біреуінің (proc типті торабы) саны. Егер кеңістіктік модельдеудің графикалық интерпретациясы қажет болмаса, онда none көрсетіледі;

p_8 - терминатордың нөмірі (типті term), оның кіріспесінде имитациялау барысында транзакциялар ағынының қарқындылығын сақтау қажет. Егер мұндай қажеттілік жоқ болса, онда none көрсетіледі;

p_9 дәлдігі (int саны 1-ден 6-ға дейін немесе none), яғни уақыт аралығына сәйкес келетін нәтижелердің шығуындағы ондық үтірден кейінгі сандар саны. Егер $p_9 = none$ болса, нәтижелер бүтін мәндер бойынша дөңгелектенеді.

Үлгі үйлестірушісін іске қосу – network функциясы. Функция network, (p_1 , p_2) модельдер бөлімшелеріндегі мәмілелерді диспетчерлендіреді, оқиғаларды біртұтас уақыт режимінде жоспарлайды және модельдің үздіксіз компоненттерін белсендіреді. Бағдарламалау тілінің синтаксисінің тұрғысынан, бұл функция құрылымдық мәлімдеме болып табылады, одан кейін модельдік графиктің сипаттамасымен блокты ашатын бұйра сызық $\{$. Бұл функция шақыру modbeg функциясы және барлық ag функциялары орындалғаннан кейін ғана жасалады. p_1 және p_2 аргументтері интегралдауды жүзеге асыратын кіші бағдарламалардың атаулары (мекен-жайлары), айырмашылық теңдеуін шешу, формулалар бойынша есептеу және т.б. Ең қарапайым модельдерде бұл функциялар қажет емес, ал dummy сөзі p_1 және p_2 параметрлері ретінде пайдаланылады. Бүкіл модельдеу кешенінің орындалуы бірінші қатеге дейін немесе modbeg функциясында p_3 аргументінде көрсетілген уақыт аяқталғанға дейін жоспарланған.

Қате өңдегіш - fault (err) функциясы. Бағанның сипаттамасының ең соңында, егер пайдаланушының назарынсыздығына байланысты транзакцияны жоқ түйінге жылжыту үшін әрекет жасалса, берілген fault (err) функциясын жазу керек. Мұндай әрекеттен кейін модельдеу тоқтайды және аяқталады err кодымен, мұндағы err - әдетте 101-ден 32 767-ге дейінгі диапазонда еркін анықталатын нөмір болып табылады. Бұл жағдайда пайдаланушыға осы қате орын алған түйін және келесі түйіннің дұрыс нөмірі деген хабарлама беріледі.

Үлгінің соңы - modend функциясы. Modend (p₁, p₂, p₃, p₄) функциясы ішкі жай-күйі поегт бұзылғаннан кейін орындалуы керек, ол модельде жалпы қателігі бар немесе modbeg мәлімдемесінде көрсетілген модельдеу уақыты аяқталғанын анықтайды. Ол компьютердің жадынан моделдеу үрдісінде жасалған басқару құрылымдарын жояды. Бұған қоса, бұл функция монитордың экранында графикалық нәтижелерді көруге және соңғы нәтижелерді есеп файлына шығаруға мүмкіндік береді. Параметр мәндері:

p₁ - модельдеу нәтижелерінің кестесі бар файлдың атауы;

p₂ - есептің бірінші бетінің нөмірі;

p₃ - әр беттегі жолдар саны;

p₄ - екі мағынасы бар: кез-келген бет, егер сіз бет файлына аударманың белгісін есеп файлына қою керек болса немесе none.

Пайдаланушының бастапқы және аяқтау функциялары. Modbeg және network функцияларының арасында модельдің таймерінде нөлдік мән болғанда, сондай-ақ кестенің түйіндерінің сипаттамасын сипаттағаннан кейін, жабылатын бұйра сызығымен аяқталған } - modend функциясынан бұрын, үлгі таймерінде соңғы мән бар болғанда - C ++ бағдарламасындағы кез-келген бағдарлама немесе C ++-та жазылған кез-келген функцияға сілтеме жасаңыз, ол негізгі модель жұмыс істегенге дейін немесе кейін жұмыс істейтін болады.

Торап сипаттамасының блогы. Торап сипаттамасы блогының құрылымы келесідей:

```
top(<n>) : [<операторлар пайдаланушы>]
          <торап түрі>(<торап параметрлері>);
[clcode < пайдаланушының операторы>]
place;
```

Тор (<n>) бастап, трансакт барлық операторларға кешіктірусіз (модель уақытының мағынасында) өтіп, келесі түйінге өту шарттары пайда болғанша place операторына енеді. Кейбір түйіндерде бір уақытта бірнеше трансакт болуы мүмкін. Олардың барлығы бірдей place мәлімдемесінде. Place-да трансактының кешігуі - бұл торапта қалу уақыты болып келеді. Түйін түрі (түйін функциясы) алдында тұрған операторлар түйінге кірер алдында трансактының орындау керек әрекеттері болып табылады. Ccode белгісін бақылайтын операторлар кіргеннен кейін орындалатын әрекеттер болып табылады. Барлық айнымалы мәндер - түйін параметрлері трансакт түйінді енгізбес бұрын анықталуы керек.

Торап түріне қарайтын операторлар трансакция құлыпталған түйінге (бос сервис немесе жеке кілт) кіруге тырысқан кезде бірнеше рет орындалуы мүмкін. Модель жұмысының барысында желілік үйлестіруші мезгіл-мезгіл «Талқылаулар» босатылған-істемейтінін тексеріп, түйіндерді окшаулап отырады. "Сауалнама" кезінде түйіннің түрін орындайтын операторлар автоматты түрде орындалады. Осылайша, осы жерде осы операторларға жаза алмайды, олардың қайтадан орындалуы модель жұмысында қате әкеледі. Ccode кейінгі операторлар тек бір рет орындалады.

Торап сипаттамасы блогында C ++ тілінде және Pilgrim командаларының кез келген жинақтарын орналастыруға болады, сондай-ақ трансакт параметрлері мен түйін күйінің параметрлерін, бағдарлама үзінділерін, C ++ тілінде жазылған функцияларын және жүйелік қоңырауларға қол жеткізуге болады. Алайда, бұл белгілі бір ережелерге сәйкес жасалуы тиіс. Біріншіден, C ++ тілінің goto үлгідегі операциясын тек бір түйінде (top белгісі функциясы мен тиісті place операторы арасында) ғана емес, сонымен қатар басқа түйінді өту үшін қолдануға болмайды. Модельдеуді аяқтауға мәжбүрлеу үшін goto орнына тек операцияны пайдалануға болады

error = коды,

онда «код» - «коды»> 100 жағдайын қанағаттандыратын пайдаланушының кез келген қате коды.

Осындай операциядан кейін бақылау координатордан таңдалады және операторға баған сипаттамалық келесі блок үшін (мысалы, функция modend) сәйкес беріледі.

Екіншіден, бірнеше орындалу мүмкіндігінің себебінен, $xd = f(x)$ немесе $x = x + 1$ типті реквизиттерді top жапсырмасы мен түйін түрі

және т.б. арасында пайдалануға тыйым салынады және маңызды компьютерлік ЭВМ процессор уақытын талап ететін ресурстық қарқынды есептеулерді жазу қажет емес. Тек тапсырма операциясын және, қажет болса, if немесе switch операциясын қолданған жөн. Мысалы:

```
top(3):if (a>0) b =4;
else b = 5;
key("Кілт",b); place;
```

Ресурстық қарқынды есептеулерді орындау қажеттілігі туындаса, кейбір айнымалылармен (мысалы, $x = f(x)$ немесе $x = x + 1$) рекурсияларды қолданыңыз, жүйелік қоңыраулармен жұмыс істемей, ресурсқа негізделген есептеу функцияларына қолжеткізіңіз, бұл түйін түрін анықтау функциясымен тиісті place құрал арқылы арнайы ccode операторы.

Түрін анықтаған оператордан кейін сіз dd пішімінде жазылған C++ тілінің кез келген мәлімдемелерінің блогын жаза аласыз

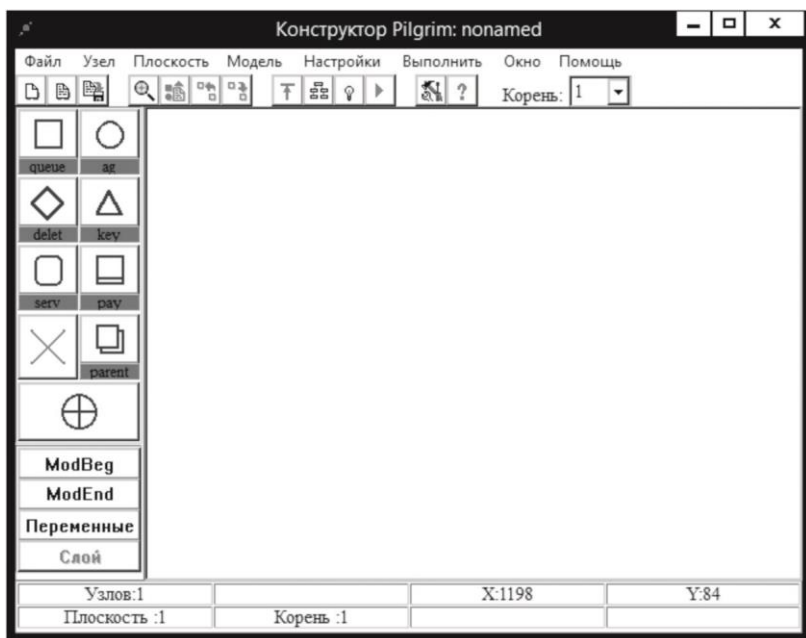
```
Ccode
{
<пайдаланушы операторлар блогы>
}
немесе жекеccodemәлімдемесі
```

9.3.

PILGRIM ИМИТАЦИЯЛЫҚ МОДЕЛЬДЕУ ПАКЕТІ

Pilgrim модельдеу жүйесінде модель жасаған кезде графикалық дизайнер Gem көмегімен модельдеу бағдарламасын жасау ыңғайлы, ол сізге үлгі диаграмма жасауға, жалпы модельдеу параметрлерін және графикалық түйін параметрлерін орнатуға және C++ үлгі кодын жасақтауға мүмкіндік береді. Болашақта бұл код Microsoft Visual Studio жүйесінде жасалған жобадан пайдаланылады. Графикалық дизайнер Gem көмегімен модельдеудің негізгі принциптерін қарастырайық.

Графикалық дизайнер басталғанда негізгі бағдарлама терезесі пайда болады (9.2-сурет). Бұл терезеде басты мәзір мен екі құралдар тақтасы бар, олардың біреуі бар



Сурет. 9.2. Графикалық конструктордың негізгі терезесі - бағандардың моделітұратын Gem.

Бұл жағдайда бір түйме түйіндердің бірнеше түріне сәйкес болуы мүмкін (үлгі параметрлерінде арнайы түйін түрі таңдалады). Мысалы, queue түймесі queue, sent және attach түйіндерін анықтауға мүмкіндік береді, ag және term түйіндерін анықтау үшін ag түймесі, delete және create түйіндерін анықтау үшін delete түймешігі, key, manager және direct түйіндерін анықтау үшін key түймешігі, ray түймешігі ray gent және down түріндегі түйіндер.

Жаңа үлгі жасау үшін Файл /Құру Жаңа мәзір пәрменін орындаңыз немесе құралдар тақтасындағы Жаңа үлгі түймешігін басыңыз. Осыдан кейін сіз модель бағандарын жасай алатын жаңа ұшақ жасай аласыз. Үлгі торабын жазықтыққа орналастыру үшін, оны құралдар тақтасынан ұшаққа дейін апару керек. Осыдан кейін, блоктың мәтінмәндік мәзіріндегі түйменің Параметрлер командасын (немесе блокты екі рет басу арқылы) пайдаланып, сіз торап параметрлерін теңшеу үшін терезені аша аласыз. Түйіндер арасындағы сілтемелерді көрсету үшін, шеңберде көк косы бар түймешікті пайдаланыңыз. Бұл түймені сілтеме шыққан түйінге

сүйреп апару керек, содан кейін осы сілтемені қамтитын түйінді басыңыз. Бағанның элементтерін жою үшін қызыл кресті бар батырманы пайдаланыңыз. Оны жойғыңыз келетін нысанға сүйреп апару керек.

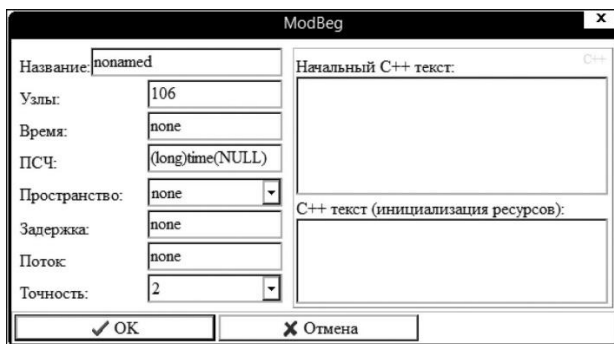
Жалпы модельдеу параметрлерін орнату үшін құралдар тақтасындағы ModBeg батырмасын пайдаланыңыз, басқан кезде параметрлері бар терезе ашылады (9.3 суретті қараңыз). Мұнда сіз модельдің атын (атау өрісі), модель уақыт бірліктеріндегі уақытты модельдеу уақытын, queue түрінің түйін нөмірі, имитациялық (Кешігу өрісі), уақыт түйінінің нөмірі, көрсетілетін кіріс ағынының динамикасы көрсетілетін кешіктірілу динамикасы көрсетіледі. Модельдеу кезінде (Ағын өрісі), алынған нәтижелерге арналған дәл сандардың саны (дәлдік өрісі) көрсетіледі және модельдік операцияның бастапқы кезеңінде (бастапқы C ++ мәтін өрісі) орындалатын C ++ пәрмендері көрсетіледі.

ModEnd батырмасын қолданып, модельдеу нәтижелері көрсетілетін файлдың атын көрсетуге және шығыс параметрлерін көрсетуге болатын терезе шақырылады.

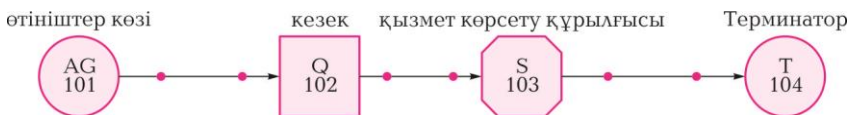
Айнымалылар түймесінің көмегімен сіз модельдеуде пайдаланылатын өзгермелі айнымалыларды анықтай аласыз және олардың түрін және бастапқы мәнін белгілей аласыз.

Gem графикалық дизайнерде жасалған графикалық модельдің мысалы 9.4-суретте көрсетілген.

Модельдік бағанды жасағаннан кейін оның дұрыстығын тексеріп, *Орындау/дұрыстығын* графикасының дұрыстығын тексеріп, C ++ файлы жасаңыз.



9.3-сурет. Үлгінің жалпы параметрлерін көрсету терезес тапсырмалары



Сурет 9.4. Gem графикалық құраушысында құрылған СМО моделі бағанының мысалы

C++ файлын іске қосу / жасау. Бұл C++ файлын Visual C++ жобасында пайдалануға болады. Жасалынған үлгі одан әрі пайдалану үшін *Файл / Сақтау* pgf-файлы ретінде сақталады.

9.4.

ИМИТАЦИЯЛЫҚ МОДЕЛЬ ДАЙЫНДАУ МЫСАЛЫ

СМО моделін дайындау үрдісін қарастырайық (екі нотариусы бар нотариалды кеңсе), олардың сипаттамалары 7.3 бөлімде аналитикалық есептелген. Ыңғайлылық үшін тағы да аталған жүйені сипаттайық.

Нотариалды кеңсе екі арналы СМО тұрады делік. Кезекті күту бөлмесіндегі орындар саны шектеулі және үшке тең. Егер күту бөлмесіндегі барлық орындар бос болмаса, онда қайта келген клиент кезекке тұрмайды. Кеңес алуға келген клиенттер ағыны қарапайым қарқындылықпен сағатына 12 клиентті құрайды. Қызмет көрсету уақыты экспоненциалды заң бойынша бөлінген және $t=7$ минутты құрайды.

Алдымен тапсырмалар талдауын орындайық және модель бағанасын құрайық. Осыдан кейін есептік талдауды және тәжірибені жүргізейік.

Аталған тапсырмада тек бір түрдегі өтініштер бар – клиенттер. Бұл өтініштер түйін көмегімен қалыптасады – транзактілер, генераторлар көмегімен. Клиенттер ағыны қарапайым болғандықтан, клиенттердің келуі арасындағы уақыт интервалын бөлу заңы экспоненциалды болып келеді және оның орташа мәні: $60/\lambda$ мин.

Генераторда қалыптастырылған транзактілер бөлігі қызмет көрсету бойынша кезекке тұрады (егер кезектегі клиенттер саны үштен аз болса), ал кейбір бөлігі бірден жүйеден шығып, қызмет көрсетілмеген өтініштер терминаторына түседі. көмекші түйінді қолдануға болады, мысалы key типі, егер tn өрісі түйін кезегінде үштен аз параметр және басқа жағдайда

қызмет көрсетілмеген транзакт терминаторы кезінде сұранымдар кезекке түседі (бұл шарттар сәйкес ауысуды таңдау кезінде кезекке шығу үшін 9.5-суретте көрсетілгендей клапан параметрлері және Ауысу шарты өрісіне беріледі).

Түйіннен кейін сұраным кезегі екі қызмет көрсету каналы бар қызмет көрсету құрылғысына түседі. Есеп шарты бойынша қызмет көрсету уақыты экспоненциалды заң бойынша 7 мин орташа қызмет көрсету уақытымен бөлінеді (бастапқы деректер бір үлгідегі уақытта беріледі). Содан кейін сұранысқа қызмет көрсету құрылғысы қызмет көрсетілген сұранымдар терминаторына түсуі қажет.

Есепке жасалған талдау негізінде Gem графикалық конструкторының көмегімен 9.6-суретте көрсетілген графикалық модельді құруға болады. Осыдан кейін ModBed (9.7-сурет) батырмасының көмегімен модельдің жалпы параметрлерін береміз, қажет болғанда ModEnd батырмасының көмегімен нәтижені шығаруға арналған параметрлерді береміз.

Свойства узла

Номер: 105

Имя: Выбор

Класс: Кей Плоскость: 1

Определить параметры...

Общий C++ текст:

До вхождения в узел После вхождения в узел

Входы: Из 101 в 105

Выходы: Из 105 в 102 Из 105 в 106

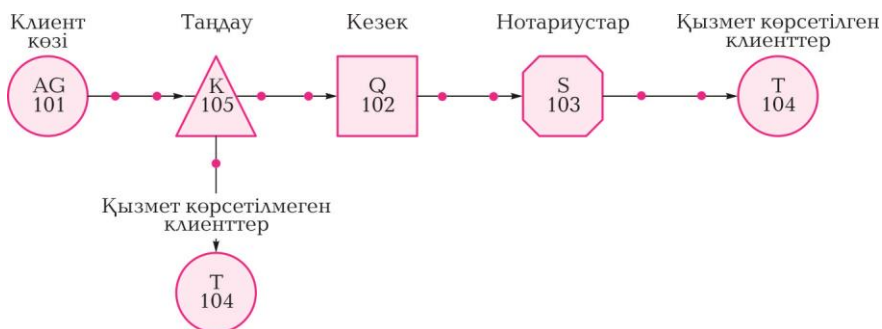
Условие перехода: addr[102]->tn < β

C++ текст:

Удалить Удалить

Ok Применить Отмена

Сур. 9.5. клапан параметрлері



Сур. 9.6. Бағандардың моделі

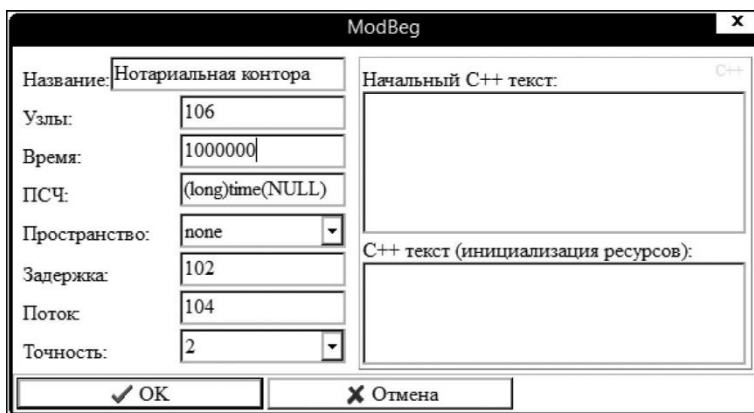


Рис. 9.7. Общие параметры модели

Содан кейін мәзір пәрменін пайдаланып, үлгі баған моделінің дұрыстығын тексереміз. Іске қосу / графиктің дұрыстығын тексеріңіз және қате болмаған жағдайда, C++ файлының Орындау/Генерациялау C++ файлының мәзір командасының көмегімен жасаймыз. Үлгіні Файл/Сақтау пәрменін пайдаланып rgf-файлында сақтау қажет, қажет болған жағдайда оны өзгертуге болады.

Төменде келтірілген C++ файлының мәтіні келтірілген.

```

#include <Pilgrim.h>
forward
{
    int fw;

```

```

modbeg("Нотариалдық контора", 107, 1000000,
(long time(NULL), none, 102, none, 104, 2);
тозақ («Клиент көзі», 101, none, expo, 60.0 / 12,
none, none, 105);
желі (күлгін, күлгін)
{
    жоғарғы (102):
    Кезек («кезек», жоқ, 103); орын;
    жоғарғы (103):
    serv («Notaries», 2, none, expo, 7.0,
none, none, 104); орын;
    жоғарғы (104):
    мерзімі («Қызмет көрсетілетін
клиенттер»); орын;
    жоғарғы (105):
    егер (addr [102] -> tn <3)
    {
    Else
    {
    fw = 106
    }
    }
key("Таңдау", fw);
place;
top(106):
term("Қызмет көрсетілмеген клиенттер");
place;
fault(123);
}
modend("pilgrim.rep", 1, 8, бет);

return 0;

}

```

Pilgrim модельдік жобасын құру және іске қосу. Модельдеу моделі Microsoft Visual C ++ ортасында жасалады. Мысал ретінде Visual Studio 2000-ні қолдана отырып, үлгіні құру процесін қарастырайық. Алдымен үлгі жобаны жасауыңыз керек. Ол үшін Visual C ++ қабықшасының File мәзірін New... ішкі мәзірін таңдаңыз, содан кейін Projects қойындысында Win32 Application терезесінің қалқымалы терезесінде таңдалады. Сол терезеде, жоба атауын және оның орналасқан жерін дискіде көрсету керек. Содан кейін ОК түймесін басу бос жобаны жасайды.

Содан кейін үлгінің мәтінін қамтитын жобаға файл қосу қажет. Бұл әрекетті орындау үшін Files қойындысының қалқымалы терезесінде C ++ Source File таңдаңыз, файл атауын енгізіңіз, Add to project құсбелгісін қойыңыз, содан кейін ОК түймешігін нұқыңыз Visual C ++ қабығының File мәзірін таңдауыңызға болады New... ішкі мәзірін таңдай аласыз. Жасалған файлда үлгінің мәтінін (мысалы, бұрын ұсынылған) немесе бұрын дайындалған файлдан көшіріп алу қажет.

Содан кейін жобаға Comctl32.lib (C: \ Program Files \ Microsoft Visual Studio \ VC98 \ Lib \ COMCTL32.LIB), Pilgrim.lib (C: \ Program Files \ Microsoft Visual Studio \ VC98 \ Lib \ PILGRIM.LIB) кітапханалары қосылған, Pilgrim.res ресурстар файлы (C: \ Program Files \ Microsoft Visual

Studio \ VC98 \ projects \ PILGRIM.RES) және тақырыптама Pilgrim.h (C: \ Program Files \ Microsoft Visual Studio \ VC98 \ Include \ Pilgrim.h) және Simulate.h (C: \ Program Files \ Microsoft Visual Studio \ VC98 \ Include \ SIMULATE.H). Осы файлдарды қосу үшін, Сіз Project мәзірі → Project To Project файлдарын таңдап, содан кейін пайда болатын терезеде қалаған файлды таңдап, ОК түймесін басыңыз. Visual Studio 2005 және 2008 үшін жоба сипаттарында (Alt + F7 деп аталатын) Configuration Properties/ Linker / Input» сипатын таңдаңыз және «libc.lib» мәтінін Ignore Specific Library өрісіне енгізіңіз

Содан кейін, модель құрастырылып, құрастырылды: Build → Rebuild All құрылымды қалпына келтіріңіз. Үлгінің мәтінінде қате болмаған жағдайда, Build → Executemәзірін пайдаланып қарамастан, қарамастан немесе Visual C ++ ортасында іске қосылатын орындалатын файл жасалады.

Үлгі басталғанда, негізгі терезе пайда болады (сурет 9.8), модельдеу үрдісі арқылы басқарылатын мәзір элементтерінің көмегімен. Төменде ең маңызды мәзір элементтері бар.

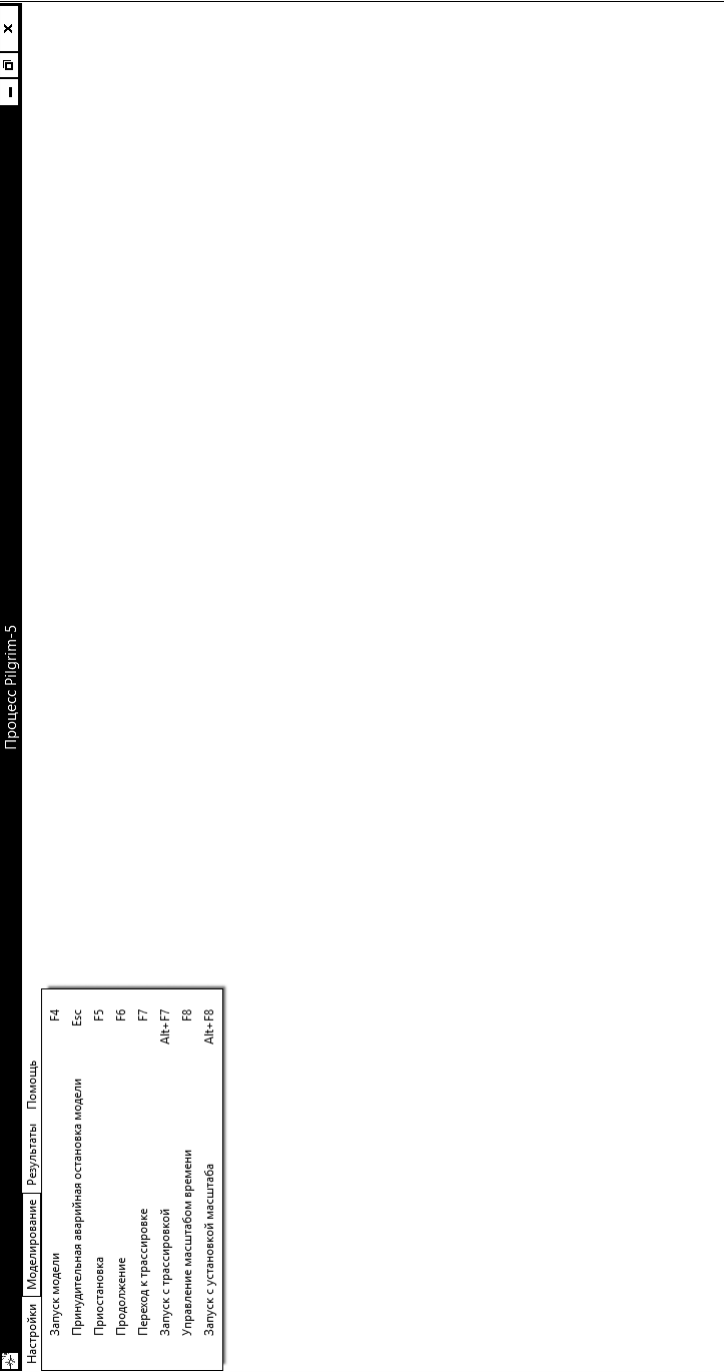
Модельдеу → Модельді іске қосу (F4) - модельдеуді бастаңыз.

Модельдеу → токтата тұру (F5) — модельдеуді тоқтату.

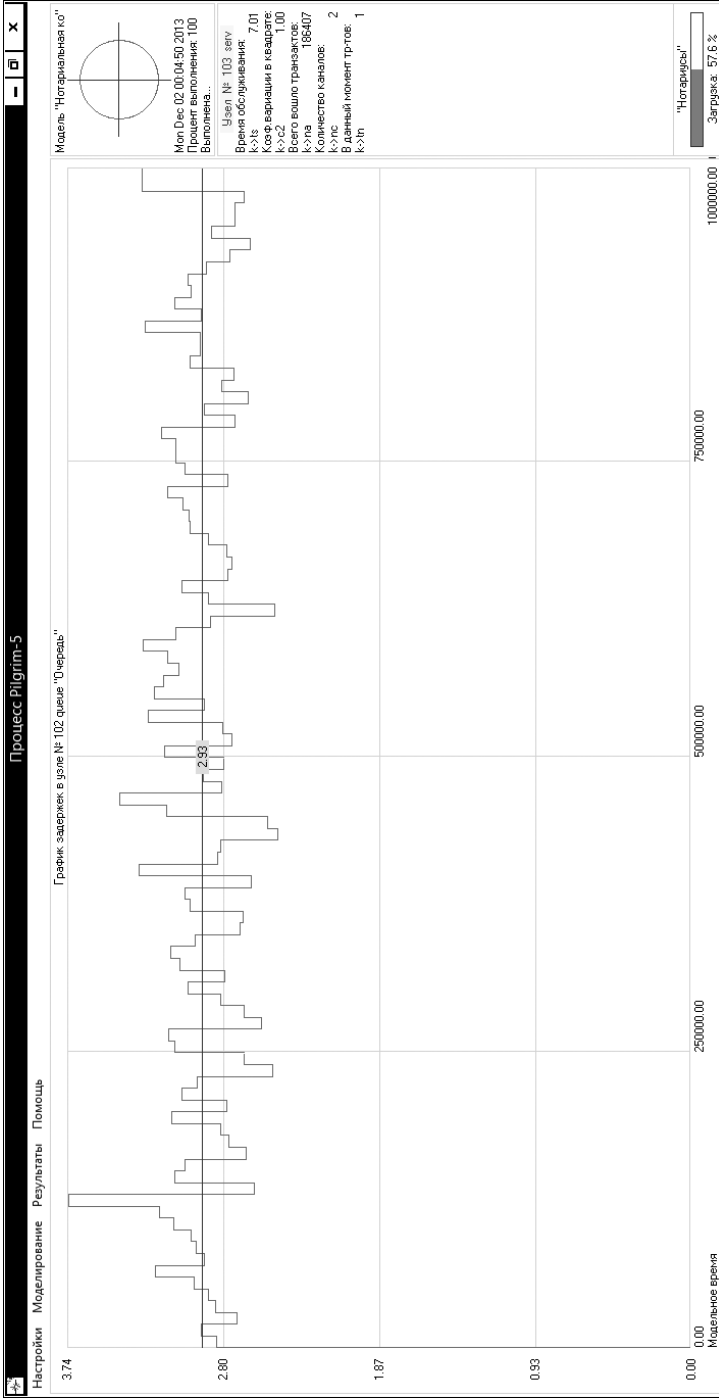
Модельдеу → Жалғасы (F6) — модельдеуді қайтару.

Нәтижелері → торап Параметрлері (F9) - параметрлер терезенің оң жағында көрсетілетін түйінді таңдаңыз.

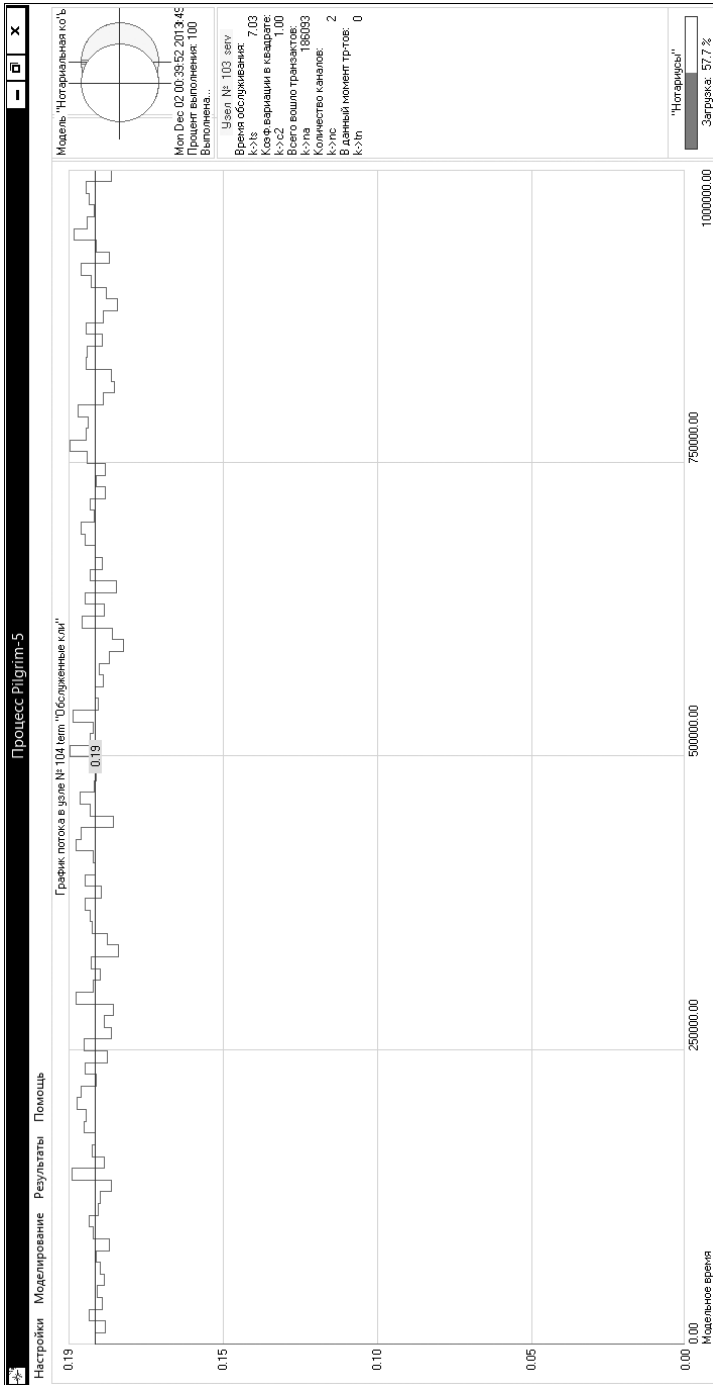
Нәтижелер → Кезектегі кешіктіру динамикасы (F10) - modbeg операторының рb параметрімен белгіленген кезекке кешігу динамикасын көрсетуге мүмкіндік береді (9.9 суретті қараңыз).



Сурет. 9.8. Pírgim моделінің терезесі



191 Сурет. 9.9. Piigrim моделiнiн гезесi



Сурет. 9.10. Рігірім моделінің терезесі

2. Жүйеде болған орташа уақыт — 9,92 мин.

3. Қызмет көрсетуден бас тарты ықтималдылығы — 0,069.

Осылайша, модельдеудің алынған нәтижелері аналитикалық есептің алдын орындалған нәтижесімен келісіледі.

Pilgrim модельдеу пакетінде құрылған модель, оның параметрлерін оңай өзгертуге мүмкіндік беретіндігін атап өткен жөн. Мысалы, қызмет көрсету каналдарының санын, кезектегі орын санын арттыруға немесе қысқартуға, клиенттің келу уақытының аралының таралу заңы мен қызмет көрсету уақыты мен олардың параметрлерінің таралу заңын өзгертуге болады. Сонымен қатар бұны С++-файлды қайта түрлендіріп, Gemграфикалық конструкторында да және сәйкес түйін функциясы параметрлерінің мәнін өзгертіп, қалыптастырылған бағдарлама кодында істеуге болады.

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

1. Имитациялық модельдеу пакеттері не үшін керек?
2. Имитациялық модельдерді жасап шығару кезінде имитациялық модельдеу пакеттері мен бағдарламалаудың әмбебап тілдерін пайдаланудың қандай артықшылықтары мен кемшіліктері бар?
3. Pilgrim имитациялық модельдеу пакеті қандай мүмкіндіктерге ие?
4. Pilgrim пакетіндегі модель қандай негізгі нысандардан тұрады?
5. Какие основные типы узлов модели в пакете Pilgrimпакетінде модель түйінінің қандай негізгі типтерін білесіз?
6. Pilgrim пакетіндегі транзакт қандай параметрлерге ие?
7. Pilgrim пакетіндегі түйін қандай параметрлерге ие?
8. Pilgrim пакетіндегі бағдарламаның жалпы құрылымы қандай?
9. Pilgrim пакетіндегі модель жұмысының нәтижесі қандай түрде шығарылады?
10. Gem графикалық конструкторы нені істеуге мүмкіндік береді?

ЗЕРТХАНАЛЫҚ ТӘЖІРИБЕ

10.1

ЖАЛҒАН КЕЗДЕЙСОҚ САНЫҢ БАЗАЛЫҚ
ГЕНЕРАТОРЛАРЫН ЗЕРТТЕУ

Тең ықтималды таратумен $[0; 1)$ аралықта бірге жалған кездейсоқ сандарды генерациялау үшін бағдарламаны (кіші) құрастыру және ретке келтіру. 10.1 кестеден нұсқаны таңдаңыз. Бұл кездейсоқ сандардың генераторының түрін, бастапқы шарттарын және т.б. көрсетеді. Берілген іріктеу көлемі және ұсақтау аумағы үшін аралық саны $[0; 1)$ жиіліктер мен статистикалық бөлу таралу функциясының гистограммасын құрастыру, математикалық күтудің бағдарламаланған бағаларын, дисперсияны, екінші және үшінші сәттерді алу. Алынған нәтижелерге талдау жасау.

Кесте 10.1. Зертханалық жұмыс № 1 нұсқалық тапсырмалар

Нұсқаның нөмірі	Тетік түрі	Бастапқы мәліметтер	Іріктеу көлемі	Ұсақтау аумағының саны
1	Мультипликативті, формула (2.4)	$Y_0 = 4\ 003, M = 4\ 096$	256	16
2	Әмбебап, формула (2.9), $\kappa = 3$	Y_k — кез келген	500	16
3	Аддитивті, формула (2.6)	$Y_1 = 3\ 215, Y_2 = 4\ 073,$ $m = 4096 \cdot 4$	5 000	10
4	Универсальный, формула (2.9), $\kappa = 1$	Y — кез келген	1 600	18

10.1 кесте жалғасы

Нұсқаның нөмірі	Тетік түрі	Бастапқы мәліметтер	Іріктеу көлемі	Ұсақтау аумағының саны
5	Квадратты конгруэнттік әдіс, формула (2.11)	Y — кез келген, $I = 12$	5 000	25
6	Ковэю квадраттыәдісі, формула (2.12)	$I = 9,$ Y шартынан $Y \bmod 4 = 2$	5 000	12
7	Квадратты конгруэнттік әдіс, формула (2.11)	Y — кез келген, $I = 12$	7 000	16
8	Макларена— Марсальи Әдісі	$\kappa = 256$	1 000	21
9	Аралас, формула (2.5)	Y — кез келген	4 000	16
10	Аддитивті, формула (2.6)	$Y = 4091, Y_2 = m - 5,$ $m = 4\ 096 \cdot 4$	1 000	16
11	Жалпыланған аддитивті, формула (2.7)	$r = 6,$..., X_6 кездейсоқ сандар кестесінен	6 000	16
12	Жалпыланған аддитивті, формула (2.7)	$r = 10,$ $x^{\wedge} \dots, x_{10}$ кездейсоқ сандар кестесінен	6 000	16
13	Аддитивный, формула (2.6)	$Y = 3\ 971, Y_2 = 1\ 013,$ $m = 4\ 096 \cdot 4$	2 000	21
14	Макларена— Марсальи Әдісі	$\kappa = 64$	2 000	16
15	Аралас, формула (2.5)	$Y = 3\ 845$	1 000	16
16	Жалпыланған аддитивті, формула (2.7)	$r = 8,$ x_j, \dots, x_8 кездейсоқ сандар кестесінен	6 000	26

Нұсқаның номеры	Тетік түрі	Бастапқы мәліметтер	Іріктеу көлемі	Ұсақтау аумағының саны
17	Макларена— Марсальи Әдісі	$k = 128$	5 000	26
18	Аралас, формула (2.5)	$Y = 4\ 001$	1500	16
19	Әмбебап, формула (2.9), $k = 2$	Y_k — кез келген	4 000	21
20	Мультипликативті, формула (2.4)	$Y_0 = 3\ 091, M = 4\ 096$	2 000	21

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 2. ЖАЛҒАН КЕЗДЕЙСОҚ САНЫҢ САПАСЫН ТЕКСЕРУ

Зертханалық жұмыста 1 жасағанда алынған нәтижелерді пайдалана отырып, Пирсон, Колмогоров өлшемдері арқылы жалған кездейсоқ сандардың сапасын тексеріңіз, сонымен қатар кестеде 10.2. көрсетілген өлшемдер:

Кесте 10.2. Зертханалық жұмыс № 2 нұсқалық тапсырмалар

Нұсқаның номеры	Өлшемі
1	Жанама белгісі бойынша сапасын тексеру
2	Серия санының өлшемі, бөлетін элемент $p = 0,25$
3	Нөлдік серия ұзындығының сынақтамасы, бөлетін элемент $p = 0,3$
4	Бірлік серия ұзындығының сынақтамасы, бөлетін элемент $p = 0,4$
5	Покер-сынақтама, $k = 2$
6	Коллекционер өлшемі

Номер варианты	Өлшемі
7	Покер-сынақтама, $k= 8$
8	Серия санының өлшемі, бөлетін элемент $p= 0,5$
9	Нөлдік серия ұзындығының сынақтамасы, бөлетін элемент $p= 0,5$
10	Бірлік серия ұзындығының сынақтамасы, бөлетін элемент $p= 0,25$
11	Коллекционер өлшемі
12	Нөлдік серия ұзындығының сынақтамасы, бөлетін элемент $p= 0,4$
13	Серия санның сынақтамасы, бөлетін элемент $p= 0,45$
14	Нөлдік серия ұзындығының сынақтамасы, бөлетін элемент $p= 0,45$
15	Бірлік серия ұзындығының сынақтамасы, бөлетін элемент $p= 0,45$
16	Серия санның сынақтамасы, бөлетін элемент $p= 0,6$
17	Покер-сынақтама, $k= 10$
18	Жанама белгісі бойынша сапасын тексеру
19	Нөлдік серия ұзындығының сынақтамасы, бөлетін элемент $p= 0,65$
20	Бірлік серия ұзындығының сынақтамасы, бөлетін $p= 0,65$

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 3. БЕЛГІЛЕНГЕН ТАРАЛУ ЗАҢЫ БАР КЕЗДЕЙСОҚ ШАМАНЫ ТҮРЛЕНДІРУ

Кесте 10.3. арқылы анықталған тапсырманың нұсқасына сәйкес кездейсоқ шамаларды түрлендіру бағдарламасын құрастыру. Іріктеу бағдарламасы бойынша алынған, жиілік гистограммасын және статистикалық тарату функциясын құрастырады, кездейсоқ шаманы математикалық күтілімі мен дисперсиясын бағалау. Эмпирикалық деректер мен теориялық таралу арасындағы сәйкестік Пирсон өлшемі немесе Колмогоров өлшемі арқылы тексеріледі. Кездейсоқ шаманың іріктеу көлемі 1 000-нан кем емес. Ұсақтау аралықтарының саны $k = 15$ немесе $k = 25$.

Кесте 10.3. Зертханалық жұмыс № 3 нұсқалық тапсырмалар

Нұсқаның нөмірі	Таралу заңы	Құрылу әдісі
1	$F(x) = \begin{cases} 0,4(x-1)^3 + 0,4, & x \in [0; 0,5); \\ 0,3x + 0,2, & x \in [0,5; 1,5); \\ 0,4(x-1)^3 + 0,6, & x \in [1,5; 2] \end{cases}$	Іріктеу әдісі
2	$F(x) = \begin{cases} \sqrt{0,25 - (x-0,5)^2}, & x \in [0; 0,5); \\ 0,3125x + 0,34375, & x \in [0,5; 1,3); \\ 1,25x - 0,875, & x \in [1,3; 1,5] \end{cases}$	Кері функци әдісі
3	$F(x) = \begin{cases} 0,25x^2, & x \in (0; 1); \\ 1,14x - 0,89, & x \in [1; 1,5); \\ 1 - 0,08(x-3)^2 , & x \in [1,5; 3] \end{cases}$	Іріктеу әдісі
4	$F(x) = \begin{cases} x, & x \in [0; 0,5); \\ 0,5, & x \in [0,5; 1); \\ 2(x-1)^2 + 0,5, & x \in [1; 1,5] \end{cases}$	Кері функци әдісі
5	$F(x) = \begin{cases} 0,3x, & x \in [0; 0,5); \\ 3x - 1,35, & x \in [0,5; 0,7); \\ 0,25x + 0,575, & x \in [0,7; 1,7] \end{cases}$	Іріктеу әдісі
6	$F(x) = \begin{cases} x^2, & x \in [0; 0,5); \\ 1,1x - 0,3, & x \in [0,5; 1); \\ 0,4x + 0,4, & x \in [1; 1,5] \end{cases}$	Кері функци әдісі

Нұсқаның нөмері	Таралу заңы	Құрылу әдісі
7	<p style="text-align: center;">Ұшбұрышты</p> $f(x) = \begin{cases} 0, & x \leq a; \\ \frac{2(x-a)}{(b-a)(c-a)}, & a < x \leq c; \\ \frac{2(b-x)}{(b-a)(b-c)}, & c < x \leq b; \\ 0, & x > b. \end{cases}$ <p style="text-align: center;">$a = 1; b = 5; c = 2$</p>	Іріктеу әдісі
8	$F(x) = \begin{cases} 0,2 \cdot 10^x - 0,2, & x \in [0; 0,3); \\ 1,5x - 0,25, & x \in [0,3; 0,7); \\ 0,25x + 0,625, & x \in [0,7; 1,5] \end{cases}$	Кері функция әдісі
9	$F(x) = \begin{cases} 0,15x, & x \in [0; 1); \\ 0,35x - 0,2, & x \in [1; 2); \\ 0,875x - 1,25, & x \in [2; 2,4); \\ 0,15x - 0,49, & x \in [2,4; 3,4] \end{cases}$	Іріктеу әдісі
10	$F(x) = \begin{cases} 0,8x^2, & x \in [0; 0,5); \\ 0,7x - 0,15, & x \in [0,5; 1); \\ 1 - e^{-0,8x}, & x \in [1; \infty) \end{cases}$	Кері функция әдісі
11	<p style="text-align: center;">Ұшбұрышты (7 тасырмаға қарау)</p> <p style="text-align: center;">$a = 0; b = 10; c = 5$</p>	Іріктеу әдісі
12	$F(x) = \begin{cases} 0,05x^3, & x \in [0; 2); \\ 1 - 2e^{-0,602x}, & x \in [2; \infty) \end{cases}$	Кері функция әдісі
13	$F(x) = \begin{cases} \sqrt{x}, & x \in [0; 0,25); \\ 0,25x + 0,4375, & x \in [0,25; 2,25] \end{cases}$	Іріктеу әдісі

Нұсканың нөмері	Таралу заңы	Құрылу әдісі
14	$F(x) = \begin{cases} 2(e-1)x, & x \in [0; 0,5); \\ e^{-2x}, & x \in [0,5; \infty) \end{cases}$	Кері функция әдісі
15	Ұшбұрышты (7 тасырмаға қарау) a = 4; b = 5; c = 4,7	Іріктеу әдісі
16	$F(x) = \begin{cases} 1 - e^{-2x}, & x \in [0; 1); \\ \frac{x - 3 + 2e^2}{2e^2}, & x \in [1; 3] \end{cases}$	Кері функция әдісі
17	$f(x) = \sqrt{R^2 - (x - a)^2};$ $R = \sqrt{\pi/2}; a = -2$	Іріктеу әдісі
18	$F(x) = \begin{cases} 0,4(x-1)^3 + 0,4, & x \in [0; 0,5); \\ 0,3x + 0,2, & x \in [0,5; 1,5); \\ 0,4(x-1)^3 + 0,6, & x \in [1,5; 2] \end{cases}$	Кері функция әдісі
19	$f(x) = \frac{\pi}{2} \sin(\pi x), \quad 0 < x \leq 1$	Іріктеу әдісі
20	Ұшбұрышты (5 тасырмаға қарау) a = -3; b = 10; c = 0	Кері функция әдісі

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 4.

ҚАЛЫПТЫ ТАРАЛУ ЗАҢЫ БАР КЕЗДЕЙСОҚ ШАМАНЫ ТҮРЛЕНДІРУ

Қалыпты таралу заңының әдісі бойынша кездейсоқ шаманы түрлендіру үшін, орталық шекті теоремаға негізделген, сонымен қатар тапсырманың нұсқасына (10.4-кесте) сәйкес анықталған әдісі арқылы бағдарлама құрастыру. Көрсетілген таралу заңының параметрі $N(m; \sigma^2)$ түрінде болады. Іріктеу бағдарламасы бойынша алынған, жиілік гистограммасын және статистикалық тарату функциясын құрастырады, кездейсоқ шаманы математикалық күтілімі мен дисперсиясын бағалау. Эмпирикалық деректер мен теориялық таралу арасындағы сәйкестік Пирсон өлшемі немесе Колмогоров өлшемі арқылы тексеріледі. Кездейсоқ шаманың іріктеу көлемі 1 000-нан кем емес. Ұсақтау аралықтарының саны $k = 15$ немесе $k = 25$

Кесте 10.4. Зертханалық жұмыс № 4 нұсқалық тапсырмалар

Нұсқа	Таралу заңы	Құрылу әдісі
1	Қалыпты, $N(3; 1)$	Аппроксимация әдісі
2	Қалыпты, $N(0; 1)$	Бокса және Малера әдісі
3	Қалыпты, $N(3; 1)$	Марсальи және Брея процедурасы
4	Қалыпты, $N(-2; 0,81)$	Аппроксимация әдісі
5	Қалыпты, $N(4.3; 0,5)$	Бокса және Малера әдісі
6	Қалыпты, $N(2; 0,9)$	Марсальи және Брея процедурасы
7	Қалыпты, $N(3.5; 0,9)$	Аппроксимация әдісі
8	Қалыпты, $N(2; 0,2)$	Бокса және Малера әдісі
9	Қалыпты, $N(-1,5; 1,7)$	Марсальи және Брея процедурасы
10	Қалыпты, $N(0; 0,1)$	Аппроксимация әдісі
11	Қалыпты, $N(2; 1)$	Бокса және Малера әдісі
12	Қалыпты, $N(-2; 1)$	Марсальи және Брея процедурасы
13	Қалыпты, $N(2; 1)$	Аппроксимация әдісі

Нұсқа	Таралу заңы	Құрылу әдісі
14	Қалыпты, $N(1; 0,7)$	Бокса және Малера әдісі
15	Қалыпты, $N(5; 1)$	Марсальи және Брея процедурасы
16	Қалыпты, $N(4,7; 0,6)$	Аппроксимация әдісі
17	Қалыпты, $N(1,5; 0,1)$	Бокса және Малера әдісі
18	Қалыпты, $N(-4; 0,1)$	Марсальи және Брея процедурасы
19	Қалыпты, $N(3; 0,1)$	Аппроксимация әдісі
20	Қалыпты, $N(2,75; 2,8)$	Бокса және Малера әдісі

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 5. ЖИИ ҚОЛДАНЫЛАТН ТАРАЛУ ЗАҢЫ БАР КЕЗДЕЙСОҚ ШАМАНЫ ТҮРЛЕНДІРУ

Бета-тарату, гамма-тарату, логарифмдік-қалыпты тарату және Вейбулла таралуына бағынатын кездейсоқ шаманы түрлендіру бағдарламаларын құрастырамыз. Іріктеу бағдарламасы бойынша алынған, жиілік гистограммасын және статистикалық тарату функциясын құрастырады, кездейсоқ шаманы математикалық күтілімі мен дисперсиясын бағалау. Эмпирикалық деректер мен теориялық таралу арасындағы сәйкестік Пирсон өлшемі немесе Колмогоров өлшемі арқылы тексеріледі. Кездейсоқ шаманың іріктеу көлемі 1 000-нан кем емес. Ұсақтау аралықтарының саны $k = 15$ немесе $k = 25$

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 6. МОНТЕ-КАРЛО ӘДІСІ БОЙЫНША МОДЕЛЬДЕУ

Тапсырманы шешуге арналған машиналық модельдеу (Монте-Карло әдісі) арқылы тапсырма нұсқасына сәйкес бағдарламаны құрастыру бағаланған ықтималдықтардың нақты мәнін қамтитын $b = 0.95$ дәйектілікпен алынған бағалау үшін сенімді аралықтарды құру. Нәтижелердің дұрыстығын тапсырманың аналитикалық шешімімен тексереді.

Нұсқаның тапсырмалары

1. Екі ракетамен қаруланған жойғыш, ауада жаулап алу мақсатында жіберіледі. Мұндай жағдайдағы жоюшының шығару ықтималдылығы, оның ішінде мақсатты шабуыл жасау мүмкіндігі, p_1 -ге тең. Егер жоюшы осы позицияға жеткізілсе, онда ол әрқайсысы бір-біріне тәуелсіз екі ракетаны босатып, ықтималдықпен p_2 ықтималдығы бар нысанның маңына алынады. Егер ракета нысананың маңайына түсірілсе, ол оны ықтималдық p_3 -ға түседі. Мақсаттың жеңіліс табу ықтималдығын бағалаңыз.

2. Атуды бір-біріне жақын бір сызықтың бойында орналасқан жанғыш бүйір бойымен екі снаряд жүзеге асырады ($k > 2$). Әрбір снаряд басқалардың тигізуіне қарамастан, бірінші бағқа ықтималдықпен p_1 түседі, екіншісі - ықтималдығы p_2 және тағы да сол сияқты. Бақты жандыру үшін бір бағқа екі соққылар немесе іргелес бағқа екі рет тигізу қажет. Бақты жандыру ықтималдығын бағалау.

3. Үш снаряд бойынша нысанаға ату жүзеге асырылады. Снарядтар бір-біріне тәуелсіз нысанаға тиеді. Әрбір снаряд үшін нысанаға тигізу ықтималдылығы p_0 тең. Егер де нысанаға бір снаряд тигізсе, ол нысанды (оны саптан шығарады) p_1 ықтималдықпен жеңіледі; егер де екі снаряд — p_2 ықтималдықпен; егер де үш снаряд — p_3 ықтималдықпен. Мақсатынан жеңілудің жалпы ықтималдылығын бағалау.

4. Екі ұшақ арасында әуе шайқасы болып жатыр: жойғыш және бомбылаушы. Атуды жойғыш бастайды: ол бомбылаушыны бір рет атады және p_1 ықтималдықпен құлатады. Егер де бомбылаушы бұл атқанда құламаса, ол жойғышты атады да және p_2 ықтималдықпен құлатады. Егер де жойғыш бұл атқанда құламаса, ол тағы да бомбылаушыны атады да және p_3 ықтималдықпен құлатады. Келесі шайқас қорытындысының ықтималдылығын бағалау:

А — бомбылаушыны құлату;

В — жойғышты құлату;

С — кем дегенде бір ұшақты құлату.

5. Натуды бір біріне тәуелсіз әрқайсысы өзінше нысанаға атуды жүргізеді. Олардың әрқайсысы шайқас қорының кпатрондары бар. Бір атқанда нысанаға тигізу ықтималдылығы, бірінші атқандағы r ($r = 1, 2, \dots, N$) тең. Өзінің нысанасына бірінші тигізгеннен бастап атқыш атуын тоқтатады. Келесі оқиғаның ықтималдылығын бағалау:

А — Барлық атқыштарда бірдей кем дегенде бір жұмсамаған патрон қалуы;

В — атқыштардың ешқайсысында барлық шайқас қоры жұмсалынбайды;

С — әйтеуір атқыштың біреуі барлық шайқас қорын жұмсайды, алқалғандары — барлығын емес.

6. Натуды кезекпен бір нысанаға атады. Ату бірінші тигізгенге дейін жалғасады. Әрбір атқыш үшін нысанаға тигізу ықтималдылығы p_i ($i = 1, 2, \dots, N$) тең. Қайсысы нысанаға бірінші тигізсе, ол жеңген болып саналады. Әрбір атқышта қорында патрон бар болады. 1-ші атқыштың жеңіске жету ықтималдығын бағалау.

7. Бомбылаушы және оның екі шабуылдаушы жойғыштарының арасында әуе шайқасы болуда. Бомбылаушы атуды бастайды; ол әрбір жойғышқа бір рет атады және оны p_1 ықтималдылығымен құлатады. Егер де бұл жойғыш құламаса, онда ол бақаның тағдырына қарамастан бомбылаушыны атады және оны p_2 ықтималдылығымен құлатады.

Келесі шайқас қорытындысының ықтималдылығын бағалау:

A — бомбылаушыны құлату;

B — екі жойғышты құлату;

C — ең болмаса бір жойғышты құлату;

D — ең болмаса бір ұшақты құлату;

E — дәл бір жойғышты құлату;

F — дәл бір ұшақты құлату.

8.

Адам, белгілі бір халық тобына жатады, қара торы p_1 ықтималдылығы болады, сарғыш шашты адам — p_2 ықтималдылығымен, ақсары — p_3 ықтималдылығымен, қызғылт сары — p_4 ықтималдылығы. Жорамалмен алты адамнан тұратын топ таңдалады. Келесі оқиғаның ықтималдылығын бағалау:

A — топтың құрамында кем дегенде төрт ақсарының болуы;

B — топтың құрамында ең болмаса қызғылт сарының болуы;

C — топтың құрамында сарғыш мен ақсарының тең болуы.

9. Құрал үш түйіннен тұрады. Құралды қосқан кезде p_1 ықтималдылығымен бірінші түйінде ақау болады, екінші түйінде — p_2 ықтималдылығы, үшінші түйінде — p_3 ықтималдылығымен. Түйіндердегі ақау бір біріне тәуеліз болады. Үш түйіннің әрқайсысы, сөзсіз құралмен жұмыс жасағанда қажет. Түйіннің істен шығуы үшін, кем дегенде екі ақау болу керек. Құрылғы пқосқанда қауіпіз тұру ықтималдылығын бағалау.

10. Ұшақтар тобының құрамында: бір басқарушы және екі жетектегіш, нысанда бомбылауға бағытталған. әрқайсысы бір бір бомбадан алып жүреді. Басқарушы ұшақ көздеуде болады, жетектегіштер — басқарушы белгі бермейінше бомбылауды жүзеге асыра алмайды. Нысанға келместен бұрын топ әуе шабуылына қарсы қорғаныстан өтеді, ондағы әрбір ұшақ бір біріне тәуелсіз p ықтималдылығымен құлатады. Егер нысанаға басқарушы ұшақ екі бірдей жетекшісімен келетін болса, олар P_{12} ықтималдылығымен нысанды қиратады. Басқарушы ұшақ бір жетекшісін алып шықса, P_{11} ықтималдылығымен нысанды қиратады. Бір бақарушы ұшақ жетекшісіз, P_{10} ықтималдылығымен нысанды қиратады. Егер де

басқарушы ұшақ құласа, онда әрбір жетекші егер де ол сақталса нысанға шығады да $P_{0,1}$ ықтималдылығымен қиратады. Нысанға қарсы әрекет ету кезіндегі толық ықтималдығын бағалау.

11. Зауыт өнім өндіреді, олардың әрқайсысында p ықтималдылықпен ақау болады. Цехта үш бақылаушы бар; өнімді тек бір бақылаушы тексеріп қарайды, бірдей ықтималдылықпен бірінші, екінші немесе үшінші. Бірінші бақылаушының ақауды табу (егер де болса) ықтималдылығы $p_i (i=1, 2, 3)$ тең. Егер де өнім цехта жарамсыз болса, ол зауыттың техникалық бақылау бөліміне (ТББ) түседі, онда егер ақау бар болса, p_0 ықтималдылықпен анықталады.

Келесі оқиғаның қтималдылығын бағалау:

A — өнім жарамыз болады;

B — өнім цехта жарамсыз болады;

C — өнім зауытта ТББ жарамсыз болады.

12. Радиолокационды станцияны нысанда бақалауға алады, кедергілері пайдаланылатын немесе пайдаланбайтын болуы мүмкін. Егер нысан кедергіні пайдаланбаса, онда бір андалымдағы танция шолуы p_0 ықтималдылығын көрсетеді; егер пайдаланса — $p_1 < p_0$ ықтималдылығымен. Айналу кезінде кедергі қолданылатындығы p ықтималдығы және қалған айналымдарда кедергі қалай қолданылатынына байланысты емес. Нысанның шолу айналым-дарында n кем дегенде бір рет көрету ықтималдылығын бағалау.

13. Топта, үш барлаушы-ұшақтардан тұратын, зымыранмен оқ жаудыру болжамымен, нысанның координаталары білу мақсатында жаудың аймағына жіберіледі. Нысанды жеңу үшін n зымырандар бөлінеді. Нысанның анықталған координаталарында бір зымыранның жеңіліс ықтималдығы p_1 , ал анықталмаған үшін — p_2 . Әрбір барлаушы нысана аймағына шығудан алдын жаудың әуе шабылуына қарсы құрылғымен құлатуы мүмкін; оның ықтималдылығы p_3 . Егер де барлаушы құлатылмаса, ол нысанның координаталарын радио арқылы хабарлайды. Барлаушының радиоаппараты p_4 сенімділігі бар. Координатаны анықтау үшін бір ғана барлаушының хабарламасы жеткілікті. Барлаушының әрекетін ескере отырып нысанның жеңілу ықтималдылығын бағалау.

14. Натқыштар ішінен төрт топты таңдап алуға болады: a_1 өте жақсы атқыш, a_2 жақсы, a_3 орташа және a_4 нашар. Бірінші топтағы атқыштың бір атқанда нысанға тигізу ықтималдылығы $P_i (i=1, 2, 3, 4)$ тең. Жорамалмен екі атқыш шақырылады, сол бір нысанаға ататын. Нысанаға ең болмаса бір рет тигізу ықтималдылығын бағалау.

15. Нысана алма мен екі сақинадан тұрады. Бір рет атқандағы алмаға тигізу ықтималдылығы p_0 тең, бірінші сақина — p_1 , ал екінші — p_2 ; нысанаға тигізе алмау ықтималдылығы p_3 . Нысанада бес рет атылыс жүзеге асырады. Екі рет алмаға тигізу және бір рет екінші сақинаға тигізу ықтималдылығын бағалау.

16. Құрылғы 10 түннен тұрады. Әрбір түйіндегі сенімділік (увқыт ішіндегі тоқтаусыз жұмыс ықтималдылығы) р тең. Түйіндер саптан бір біріне тәуелсіз шығады. Уақыт ішіндегі ықти-малдылығын бағала: а) кем дегенде бір түйіннің қабылдамауы; б) дәл бір түйіннің қабылдамауы; в) дәл екі түйіннің қабылдамауы; г) кем дегенде екі түйіннің қабылдамауы.

17. Мақсаты бойынша төрт тәуелсіз атыс жүргізіледі. Әр атыс кезіндегі тигізу ықтималдылығы әр түрлі және тең: р1, р2, р3, р4. Ықтималдылықты бағалау Р04; Р14; Р24; Р34; Р44 бірде бір, бір, екі, үш, төрт рет тигізу; ең дегенде бір рет тигізу R14 ықтималдылығы; кем дегенде екі рет тигізу R24 ықтималдылығы.

18. Зауыт өнім өндіреді, олардың әрқайсысы сандық төрт түрімен тартылады. Бірінші сынақ р1 ықтималдықпен өнімді қауіпсіз түрде өтеді; екінші — р2 ықтималдықпен; үшінші — р3 ықтимал-дықпен және төртінші — р4 ықтималдықпен. Өнімді жақсы өткізу ықтималдығын бағалау:

А — барлық төрт сынақ;

В — дәл екі сынақ (төртеуінің ішінен);

С — кем дегенде екі сынақ (төртеуінің ішінен).

19.

Зауыт өнім өндіреді, олардың әрқайсысы г (бір біріне тәуелсіз) ықтималдықпен ақауының бар болуы. Ақауды тексеру кезінде, егер де ол болса р ықтималдықпен анықталады. Бұйымды бақылау кезінде зауыттан п өнім таңдалады. Келесі оқиғаның ықтималдылығын бағалау:

А — ешбір өнімде ақаулық болмайды;

В — өнімнің ішінен дәл екі ақаудың болуы;

С — өнімнің ішінен кем дегенде екі ақаудың болуы;

20. Ұшақ оқ жаудырады п тәуелсіз атыстармен; атыстың әр қайсысы р1 ықтималдылығымен аймаққа тигізеді, ол ұшақты жедел түрде жеңіліс тапқызады; жанармай шанына р2 ықтималдықпен тигізіледі және ұшаққа р3 ықтималдықпен тигізеді. Снаряд, жанармай шанына тиген, онда сағатына жанармай клистрі ағып жатқан тесік қалдырады. Жанармай М литрін жоғалтқан ұшақ шайқасқа жарамсыз болып қалады. Ұшаққа оқ жаудырғаннан бір сағаттан кейін шайқасқа жарамсыз болу ықтималдылығын бағалау.

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 6. **КЕЗДЕЙСОҚ КЕЗУІ МОДЕЛЬДЕУ**

Әрбір тапсырманың бағдарламалық өндеуді кездейсоқ кезуі имитациялық модельін көздейді, оның көмегімен қажетті нәтижелер алуға болады. Тәжірибелердің белгілі бір санын жүргізу нәтижесінде зерттелетін параметрдің статистикалық таралуын (гистограмма және эмпирикалық

бөлу функциясы) құру және белгілі заңдардың бірі (қалыпты, экспоненциалды, логорифмдік-қалыпты және т.б.) арқылы алынған таратуды мақсаттылық жуықтамасы анықтайды.

Нұсқаның тапсырмасы

1. Бір өлшемді кездейсоқ кезуі. $M = 20$ қадам үшін жаяу жүргіншілерге кететін қашықтығын анықтайтын модельін жасаңыз.

2. Екі өлшемді кездейсоқ кезуі. $M = 10$ қадам үшін жаяу жүргіншілерге кететін қашықтықты анықтайтын модельін жасаңыз.

3. Сіңіретін экранда қарапайым кездейсоқ кезуі. Кезуі уақытын анықтау үшін машиналық модельін жасаңыз.

4. «Аралар» квадраттық торда. «Омарта» N -нен «Ара» бастапқыда координаттар басында ортасымен бір шеңберде орналасқан. Әрбір кезеңде әрбір «ара» төрт бағыттың бірінде кездейсоқ түрде жылжиды: солтүстікке, оңтүстікке, шығысқа және батысқа. $M = 8$ қадам үшін жаяу жүргіншілерге кететін қашықтықты анықтайтын модельін жасаңыз. Әрбір уақытша аралықта

ішінде «Аралар» бірлік ұзындығының қадамын жасайды. Орташа есеп N «Аралар» бойынша орындалады.

5. Ұшбұрышты торда кезуі. Ұшбұрышты торда «Аралар» кездейсоқ кезуі имитациялық модельін жасаңыз. Уақыт бойынша әрбір қадамда «Ара» алты ықтимал бағыттардың бірінде теңдей ауысады. $M = 8$ қадам үшін «Ара» қандай қашықтыққа кетеді?

6. Жауын тамшыларының түсу моделі. Жеңіл желдің кездейсоқ екпінің әсерінің арқасында, жаңбыр тамшыларының түсуін шаршы торда кездейсоқ кезуін модельдеуге болады (сур. қар. 6.9). Қозғалыс көлденең сызықтан (жер үстінен) h қашықтықта орналасқан түйінінен басталады. Сонымен қатар төменгі қадам r_1 ықтималдылығы жоғарғы қадам r^{\wedge} ықтималдылығы үлкенірек. Секіру ықтималдылығын бірдей таңдаған абзал: $r_1 = 0,5$; $r_1 = 0,1$; $r^{\wedge} = r^{\wedge} = 0,2$.

Тамшылаудың көлденең сызыққа жететін уақытын және t -ның h функциясына тәуелділігін (4 ... 6 мәндері) анықтаңыз.

7. Кездейсоқ кезуді шектеу. Сіңіретін экранда қарапайым кездейсоқ кезуі тапсырмасы келесі түрлендірулер түрінде қарастырылуы мүмкін. Бір өлшемді торда нүктелерде сіңіретін түйіндері (тұзақ) бар $x = 0$ және $x = a$ ($a > 0$). Бөлшек x_0 ($0 < x_0 < a$) нүктесінен қозғалысты бастайды және бірдей ықтималдылықпен жақын көршілес түйіндерге өтеді. Бөлшектің оның сіңуіне дейін өту уақытын анықтаңыз.

8. Жасушада кезуі. Жасушаларда имитациялық кездейсоқ кезуі модельін жасаңыз. Уақыт бойынша әрбір қадамда аралар үш бағыттың біреуінде бірдей ықтималдылықта қозғалады. $M = 8$ қадам үшін «ара» қанша қашықтыққа кетеді?

9. Үш өлшемді торда кездейсоқ кезуі. Бөлшектің кететін қашықтықты бағалау, бұл үш өлшемді торда кезуі бірдей ықтималды

болады. Кезуі қадамдардың саны $M = 10$. Зерттеліп жатқан процеспен қатар, барлық үш координаттар үшін бастапқыдан кететін қашықтықты бөлек анықтау.

10. Персистентті кездейсоқ кезуі. Персистентті кездейсоқ кезуі өту немесе «секіру» ықтималдығы соңғы өтуге байланысты. Персистентті бір өлшемді кездейсоқ кезуі жағдайда, қадамдар тек жақын көршілес түйіндерде жасалады. $k - 1$ қадам жасады деп ойлайық. Содан соң k -й қадам жасайды сол бағытта а ықтималдылығымен, а қарама-қарсы бағытта $1 -$ а ықтималдылығымен қадам жасайды. \neg

$M = 8$ қадамын онда $a = 0,2$ и $a = 0,4$ бастапқы жағдайдан бөлшектің орташа кетуін анықтау қажет.

11. Айнымалы қадаммен кездейсоқ кезуі. Барлық рұқсат етілген секіру ұзындығы бар бір өлшемді кездейсоқ кезуді қарастырыңыз. Ықтималдылығы, қадам ұзындығы j тең, $P(j) = \exp(-j)$ түрде болады. Бастапқы күйінен 10 қадамнан кейін қашықтыққа кететінін анықтау.

12. Айнымалы қадаммен тор. Кездейсоқ кезуін 11 тапсырмадағыдай қарастырамыз, қадам ұзындығының таралу ықтималдылығы $P(j) = a j^2$. Бұл жерде $a = 6/p^2$. Бастапқы күйінен сегіз қадамнан кейін қашықтыққа кететінін анықтау.

13. Үздіксіз кездейсоқ кезуі. Кез-келген кездейсоқ кезудің алғашқы үздіксіз модельдерінің бірі Рейлидің 1919 жылы ұсынған болатын. Рейли үлгісінде әр қадамның ұзындығы - а ықтималдық тығыздығы $p(a)$ және әрбір қадамның кездейсоқ бағытымен бөлінген кездейсоқ айнымалы.

Ықтималдылығының тығыздығы $p(a)$ $0,5$ -тен $1,5$ -ке дейінгі диапазонда теңдестірілсе және қозғалыс бағыты 1° дәлдікпен бірдей таңдалса, жаяу жүргіншінің бастау нүктесінен бес қадам кеткенін анықтау.

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 6. ЖҮЙЕНІҢ ЖАППАЙ ҚЫЗМЕТІН МОДЕЛЬДЕУ

ЖЖҚ модельдеу бағдарламасын құру, нұсқа тапсырмаларымен анықталған. ЖЖҚ тиімді функциялаудың негізгі сипаттамасын бағалау. Нәтижелерін нақты мағынасымен салыстыру, тарау 7 келтірілген формула бойынша аналитикалық есептеу кезінде алынған.

Нұсқа тапсырмалары

1. Бір арналы ЖЖҚ бас тартуы бірыңғай телефон желісін көрсетеді. Жолдың бос емес уақытына келген тапсырыс (қоңырау) бас тартылды. Оқиғалардың барлық ағындары ең қарапайым. Ағынның қарқындылығы минутына $X = 0,95$ шақыру болып табылады. Әңгіменің орташа ұзақтығы $t = 1$ мин. Жұмыс тәртібі орнатылған ЖЖҚ ықтималдық сипаттамаларын анықтаңыз.

2. Бір арналы ЖЖҚ бас тартуы минутына $X = 0,5$ сұрау қарқындылығы бар тапсырыстардың қарапайым ағымын алады. Тапсырыстың қызмет

көрсету уақыты $t = 1.5$ мин. Жұмыс тәртібі орнатылған ЖЖҚ ықтималдық сипаттамаларын анықтаңыз.

3.Компьютер орталығында бес дербес компьютер бар. Есептеу орталығына түсетін тапсырыстардың қарапайым ағымы сағатына $X = 10$ тапсырмасының қарқындылығына ие. Тапсырманы шешудің орташа уақыты - 12 минут. Барлық компьютерлер бос болмаса, тапсырыстан бас тартылады. Қызмет көрсету жүйесінің ықтималдық сипаттамаларын (есептеу орталығы) табыңыз.

4.Аудиторлық ұйымда тәулігіне $X = 1,5$ тапсырыс қарқындылығы бар қызметтерге қарапайым сұраныс ағыны алады. Қызмет мерзімі көрсетілген заң бойынша бөлінеді және орта есеппен үш күнге тең болады. Аудиторлық ұйымда аудитті жүзеге асыратын бес тәуелсіз бухгалтер бар (тапсырыс қызметі). Тапсырыстың кезегі шектелмейді. Аудиторлық ұйымның стационарлық тәртібінде жұмыс істейтін ЖЖҚ ретінде ықтималдық сипаттамаларын анықтаңыз.

5.Техбайқау пунктінде сағатына $X = 4$ қарқындылығы бар қарапайым тапсырыстар (автомобильдердің) ағынына ие болады. Байқау уақыты көрсетілген заңға сәйкес бөлу және орташа есеппен 17 минутқа тең, кезекте тұрған автокөлік бесеуден аспау керек. Техбайқау пунктінiң ықтималдық сипаттамаларын орнатылған тәртіпте анықтаңыз.

6.Кәсіпорынның бухгалтериясында кассирлер бар, олардың әрқайсысы сағатына орта есеппен 30 адам жұмыс істей алады. Жалақы алатын қызметкерлердің ағымы - қарапайым, қарқынды, сағатына 40 қызметкерге тең. Кассалық бөлімде кезек шектелмеген. Қызмет көрсету уақыты экспоненталық бөлу туралы заңға бағынады. ЖЖҚ-ның ықтималдық сипаттамаларын стационарлық тәртіппен есептеп, осындай өнімділікпен алғашқы екі жұмыс істейтін болса кәсіпорын үшін үшінші кассирді қабылдаудың мақсаттылығын анықтаңыз.

7.Жинау цехының құрастыру бөлімінде үш қойма қызметкері жұмыс істейді. Орта есеппен 1 минутта 0,8 жұмысшы ($X = 0,8$) құрылғы келеді. Бір жұмысшыға қызмет көрсету қоймашының $t = 1,0$ минутын ұстайды. Кезекте шектеу жоқ. Құралдың артында кезекте жұмысшылардың ағымы Пуассон болып табылады, және қызмет көрсету уақыты экспоненталық бөлу заңына бағынады. Қызметкердің жұмысының 1 минутының құны - 30 д.е., ал қойма қызметкері - 15 д.е. Стационарлық жұмыс тәртібінде аспаптық бөлімшедегі (қызмет көрсету уақытының құны)

Осы қызмет ұйымы үшін семинардың орташа шығындарын табыңыз.

8.Билеттер кассасы үздіксіз жұмыс істейді. Билетті бір кассир сатады. Орташа қызмет көрсету уақыты — 2 мин әрбір адамға. Жолаушылардың орташа саны, бір сағат ішінде билет кассасында билеттерді сатып алғысы келетін, сағатына $X = 20$ жолаушы. Жүйенің барлық ағындары қарапайым болып табылады. Стационарлық жағдайындағы жұмыс тәртібі ЖЖҚ сипаттамаларын анықтаңыз.

9.Автокөлікті диагностикалау қызметі - бұл бір арналы ЖЖҚ бас тартуға ұшыраған. Қызмет орны бос емес кезде қабылданған диагностикаға арналған тапсырыстар қабылданбайды. Диагностикаға арналған аппараттардың ағынының қарқындылығы сағатына $X = 0,5$ автомобиль. Диагностиканың орташа ұзақтығы $t = 1,2$ сағат. Жүйенің барлық оқиғалары қарапайым болып табылады. Жүйенің орнатылған тәртібінің сипаттамаларын анықтаңыз.

10.Автоқұю станциясы ЖЖҚ бір арнада қызмет ететін және бір бағанмен қамтиды. АЖҚС-дағы аумағында бір мезгілде үш автокөлікке жанармай құю үшін кезекте тұруға рұқсат етілген. Кезекте кез-келген үш автокөлік бар болса, станцияға келетін тағы бір автокөлік, бірақ айналып өтетін болады. Май құюға келген автокөліктердің қозғалысы минутына $X = 0.7$ автомобиль қарқындылығына ие. Құю процесі орташа алғанда 1,25 минутты құрайды. Барлық ағындар қарапайым. Жүйенің орнатылған тәртібінің ЖЖҚ сипаттамаларын анықтаңыз.

11.Темір жолды сұрыптау тау-кен құрамдары келеді сағатына $X = 2$ қарқындылығы бар. Тау-кен құрамының қызмет көрсететін орташа уақыт 0,4 сағат. Сол уақытта келген құрамдар, тау-кен бос болмағанда, кезекке тұрып, және келу паркінде күтіледі, кез-келген бір құрамды күтуге болатын үш қосалқы жүру жолы бар. Сол уақытта келген құрамдар, барлық үш қосалқы жүру жолы келу паркінде бос емес болса, сыртта осы жолға кезекке тұрады. Оқиғалардың барлық ағындары ең қарапайым. Жүйенің орнатылған тәртібі ЖЖҚ сипаттамаларын анықтаңыз.

12.АЖҚС станциясының жұмысы қарастырылған, онда үш толтыру бағандары бар. Бір автокөлікке құю орташа алғанда 3 минутты құрайды. Орташа алғанда, жанармай құю бекетінде әрбір минут сайын көлік жанармай құюға мұқтаж. Кезекте орын саны шектеусіз. Автожанармай құюға кезекте тұрған барлық автокөліктер өздерінің кезектерін күтеді. Жүйенің барлық ағындары қарапайым болып табылады. Стационарлық режимде жанармай құю станциясының жұмысының ықтималдық сипаттамаларын анықтаңыз.

13.Автокөліктің техникалық қызмет станциясына орта есеппен екі сағат сайын бір көлік келеді. Станцияда алты жөндеу бекеті бар. Қызметті күткен автомобильдердің кезегі шексіз. Бір машинаның орташа қызмет көрсету уақыты - 2 сағат. Жүйенің барлық ағындары ең қарапайым. Автомобильді техникалық қызмет көрсету станциясының сипаттамаларын анықтаңыз.

14.Екі арналы қарапайым ЖЖҚ бар бас тартуға ұшыраған. Оның тапсырыс сағатына $X = 3$ қарқындылығы бар оған кіру ағынын алады. Тапсырысты бір рет қолдануға арналған орташа жұмыс уақыты $t = 0,5$ сағат. Әр тапсырыста 5 ш.б. кіріс пайда болады. Арнаның мазмұны - 3 сағ/ш.б өтеді. ЖЖҚ арналарының санын үшке дейін ұлғайтудың экономикалық жағынан тиімді екендігін анықтаңыз.

15. Дүкенде сағатына орташа 30 тұтынушыға қызмет көрсете алатын бір сатушы жұмыс істейді. Сатып алушылардың ағыны қарқындылығы сағатына 60 тұтынушыға тең қарапайым. Барлық сатып алушылар «шыдамсыз» және кезек кезегінде бес адам болған жағдайда кетеді (оларға қызмет көрсететіндерге қосымша). Оқиғалардың барлық ағындары ең қарапайым. Тұрақты жұмыс істеу үшін дүкеннің сипаттамаларын анықтаңыз.

16. Телефон аппараттарының тоғыз жұмыс арнасымен кіруі сағатына орташа есеппен 120 тапсырыс қабылдайды. Барлық арналар бос болмаса, тапсырыстан бас тартылады. Бір арнадағы орташа қызмет көрсету уақыты - 4 минут. Жүйенің барлық ағындары қарапайым болып табылады. Телефон станцияларының сипаттамаларын анықтаңыз.

17. АЖҚС станциясының жұмысы қарастырылған, онда үш толтыру бағандары бар. Бір автокөлікке құю орташа алғанда 4 минутты құрайды. Орташа алғанда, жанармай құю бекетінде әрбір минут сайын көлік жанармай құюға мұқтаж. Кезекте орын саны шектеусіз. Автожанармай құюға кезекте тұрған барлық автокөліктер өздерінің кезектерін күтеді. Жүйенің барлық ағындары қарапайым болып табылады. Стационарлық режимде жанармай құю станциясының жұмысының ықтималдық сипаттамаларын анықтаңыз.

18. Кезек шарты үш орынмен қарапайым бір арналы ЖЖҚ үшін ықтималдық сипаттамаларын есептеңіз, $X = 4$ тапсырыс бойынша сағатына, $t = 0,5$ сағ. Егер де кезек орындардың төртке дейін көбейтсек, осы сипаттамалардың қалай өзгертінін біліңіз.

19. Есептеуге арналған тапсырмалар қабылдайтын бір арналы ЖЖҚ - ЭЕМ. Тапсырыстар ағыны ең қарапайым болып табылады, бұл тапсырыстар арасындағы орташа уақыт аралығы $t = 10$ минут. Қызмет мерзімі экспоненталық заңға сәйкес, $\lambda = 8$ минуттың математикалық күтуімен бөлінеді. ЖЖҚ-дағы тапсырыстардың орташа санын, кезектегі тапсырыстың орташа саны, жүйеде және кезекте тапсырыстардың қалыпты уақытын анықтау.

20. Жүйенің жалпы қызметін — үш терезеден тұратын билет кассасы (үш кассирмен) және шектелмеген кезекпен. Билетті сатып алғысы келетін жолаушылар, 20 минут сайын орташа есеппен бес адам келеді. Жолаушылар ағымы қарапайым деп санауға болады. Кассир орташа есеппен 10 минут ішінде үш жолаушыға қызмет көрсетеді. Қызмет көрсету уақыты экспоненталық бөлу туралы заңға бағынады. Стационарлы жағдайындағы жұмыс тәртібі ЖЖҚ сипаттамаларын анықтаңыз.

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 6.

ЭКСПЕРИМЕНТТІҢ АЙЛАЛЫ ЖОСПАРЛАУ

Әрбір тапсырма кездейсоқ кезуі бағдарламалық өңдеуді модельдеуді

дамытуды көздейді, оның көмегімен зертханалық жұмыстың екі бөлігіне қажетті нәтижелер алына алады. Жұмыстың бірінші бөлігінде көрсетілген параметрдің орташа мәнін бағалау үшін эксперименттің нәтижелері жоспарланып, қабылданады. Екінші бөлімде белгілі бір дәлдікпен берілген параметрдің үлгі талғамын бағалау үшін эксперименттің нәтижелерін жоспарлайды және алады. Сынамалық эксперимент жүргізу барысында жоспарлау кезінде параметрдің статистикалық таралуы жасалады және қалыпты аппроксимациялық мақсаттылық таралу осы заңымен анықталады. Осыған байланысты эксперименттің көлемін анықтау үшін таңдалады. Нұсқаның тапсырмасы Зертханалық жұмыс № 7 алынады.

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 6. **PILGRIM МОДЕЛЬДЕУ ҚҰРАЛ ТӘСІЛІМЕН** **ЖҮЙІНІҢ ЖАЛПЫ ҚЫЗМЕТІН МОДЕЛЬДЕУ**

Pilgrim модельдеу құрал тәсілдерін пайдалану көмегімен № 8 зертханалық жұмыстың белгілі бір нұсқа тапсырмаларымен ЖҚЖ модельдеуді орындау. Модельдеу нәтижесін № 8 зертханалық жұмыс алынған нәтижелермен салыстыру..

1. Бережная Е.В. Математические методы моделирования экономических систем : оқу құралы / Е.В.Бережная, В. И. Бережной. – М. : Қаржы және статистика, 2002.
2. Варфоломеев В.И. Алгоритмическое моделирование элементов экономических систем: Практикум : оқу құралы / В. И. Варфоломеев. – М. : Қаржы және статистика, 2000.
3. Емельянов А.А. Имитационное моделирование экономических процессов : оқу құралы / А. А. Емельянов, Е.А.Власова, Р. В.Дума. – М. : Қаржы және статистика, 2002.
4. Золотарев В.В. Компьютерное моделирование : оқу құралы/ В.В. Золотарев, Г. В. Овечкин, П. В. Овечкин. – Рязань : РГРТУ, 2008.
5. Кельтон В. Имитационное моделирование. Классика CS/ В. Кельтон, А.Лоу. – 3-ші басылым. – СПб. : Питер ; Киев : BHV, 2004.
6. Кнут Д. Искусство программирования. Том 2. Получисленные алгоритмы / Д. Кнут. – 3-ші басылым. – М. : Вильямс, 2007.
7. Лабскер Л.Г. Теория массового обслуживания в экономической сфере : оқу құралы / Л.Г.Лабскер, Л. О.Бабешко. – М. : Банктер және биржа ; ЮНИТИ, 1998.
8. Нейлор Т. Машинные имитационные эксперименты с моделями экономических систем / Т. Нейлор. – М. : Мир, 1975.
9. Основы компьютерного моделирования систем / Д. Е. Артемкин, В. В. Баринов, Г. В. Овечкин, И. М. Степнов ; А. Н. Пылькин редакциясымен. – М. : Бином. Білімдер зертханасы, 2004.
10. Пузикова Л.А. Основы имитационного моделирования на ЭВМ. Методические указания к лабораторным работам / Л. А. Пузикова. – Рязань : РГРТУ, 1991.
11. Рыжиков Ю.И. Имитационное моделирование. Теория және технология / Ю. И. Рыжиков. – СПб. : КОРОНА принт ; М. : Альтекс-А, 2004.
12. Советов Б.Я. Моделирование систем : оқулық / Б.Я.Советов, С. А.Яковлев. – М. : Жоғарғы мектебі, 2001.
13. Шеннон Р. Имитационное моделирование систем: наука и искусство / Р. Шеннон. – М. : Әлем, 1978.

Құрметті оқырман!	3
Алғы сөз	4
Тарау 1. Компьютердің модельдеудің жалпы түсінігі	7
1.1. Модель және модельдеу түсінігі	7
1.2. Модельдеу түрлерін жіктеу	9
1.3. Имитациялық модельдеу	11
1.4. Компьютердің модельдерді құры мен пайдаланудың негізгі сатылары	14
Тарау2. Жалған кездейсоқ санның базалық генераторларын бағдарламалау 17	17
2.1. Жалпы мәлімет	17
2.2. Кездейсоқ санның арифметикалық генераторлары	20
2.2.1. Конгруэнтті (сызықты) әдіс	20
2.2.2. Кездейсоқ сан генераторларын біріктіру	26
2.2.3. Сызықсыз формула негізіндегі алгоритмдер	30
Тарау3. Кездейсоқ сан генераторларын тексеру әдісі	32
3.1. Тең ықтимал таралу жиілігінің гистограммасы мен статистикалық функциясы	32
3.2. Таралу көрсеткіштерінің статистикалық бағасы	34
3.3. Жазықтықта таралуы	39
3.4. Пирсонның χ^2 критеріі	41
3.5. Колмогоров критеріі	46
3.6. Топтама критеріі	48
3.7. Жанама белгілер бойынша біркелкілікті тексеру	53
3.8. Біріктірілген тесттер	54
3.8.1. Покер-тест	55
3.8.2. Коллекция жиюшы критеріі	57
Тарау4. Берілген таралу заңы бар кездейсоқ шаманы модельдеу әдісі	60
4.1. Кері функция әдісі	60
4.2. Кесек-сызықты жуықтау әдісі	64
4.3. Эмпирикалық деректері бойынша кездейсоқ шаманы модельдеу	66
4.4. Таңдау әдісі	69
4.5. Кездейсоқ шаманың қалыпты таралуын түрлендіру	72
4.5.1. Жуықтау әдісі	74
4.5.2. Орталық шекті теореманы пайдалану	75
4.5.3. Бокс және Маллер әдісі	77
4.5.4. Марсаль мен Брей әдісі	78
4.6. Арнайы таралу заңы бар кездейсоқ шаманы түрлендіру	80
4.6.1. Экспоненциальды таралуды модельдеу	80
4.6.2. Бета-таралуды модельдеу	81
4.6.3. Гамма-таралуды модельдеу	84

4.6.4.	Логарифмді-калыпты тарау заңын модельдеу.....	86
4.6.5.	Моделирование распределения Вейбулла.....	88
Тарау 5. Дискретті уақиға мен таралуды модельдеу.....		91
5.1.	Туынды дискретті таралуды модельдеу	91
5.2.	Бернуллі таралуын модельдеу.....	93
5.3.	Биномиальды таралуды модельдеу	94
5.4.	Геометриялық таралуы бар кездейсоқ шаманы модельдеу.....	97
5.5.	Пуассон таралуын модельдеу.....	98
5.6.	Қарапайым уақиғаны модельдеу	100
5.7.	Үйлеспейтін уақиғаның толық тобын модельдеу.....	102
5.8.	Күрделі уақиғаны модельдеу	104
Тарау 6. Статистикалық сынақ әдісінің көмегімен модельдеу.....		108
6.1.	Монте-Карло әдісі	108
6.2.	Бір өлшемді кездейсоқ кезу	116
6.3.	Екі өлшемді кездейсоқ кезу	117
6.4.	Сіңіретін экрандары бар қарапайым кездейсоқ кезу	119
6.5.	Жанбыр тамшысының түсу моделі	120
6.6.	Персистентті кездейсоқ кезу	121
Тарау 7. Жаппай қызмет көрсету жүйелерін модельдеу.....		124
7.1.	Жаппай қызмет көрсету жүйелерінің негізгі сипаттамалары	124
7.2.	Бір қызмет көрсету құрылғысы бар жүйе	128
7.3.	Жаппай қызмет көрсететін көп каналды жүйе	131
7.4.	Жаппай қызмет көрсететін жабық жүйе	135
7.5.	Жаппай қызмет көрсететін жүйені компьютерлік модельдеу.....	140
Тарау 8. Компьютерлік тәжірибені жоспарлау		149
8.1.	Жалпы түсінік	149
8.2.	Тәжірибені стратегиялық жоспарлау	152
8.3.	Компьютерлік тәжірибені тактикалық жоспарлау.....	153
8.3.1.	Орнатылған тәртіптің бастапқы шарты және олардың қол жетуге әсері.....	154
8.3.2.	Таңдамалы жиынтықтың орташа мәнін бағалау.....	155
8.3.3.	Чебышев теоремасын қолдану.....	157
8.3.4.	Пайыздық қатынасты бағалау.....	158
8.3.5.	Жиынтық дисперсиясын бағалау.....	160
8.3.6.	Автокорреляцияланған деректер.....	161
Тарау 9. Имитациялық модельдеу пакеттері.....		163
9.1.	Имитациялық модельдеу пакеттерінің тағайындалуы	163
9.2.	Pilgrim имитациялық модельдеу пакеттері.....	164
9.2.1.	Модельдің негізгі нысандары.....	165
9.2.2.	Имитациялық модель түйіндерінің типі.....	167

9.2.3.	Түйінді басқару командалары.....	173
9.2.4.	Транзакт параметрлері және түйін параметрлерінің күйі.....	174
9.2.5.	Модельдеудің кестелік нәтижелері.....	176
9.2.6.	Модельді сипаттау тілі.....	178
9.3.	Gen графикалық конструкторы.....	182
9.4.	Имитациялық модельді әзірлеу мысалы.....	185
Тарау 10. Зертханалық тәжірибе.....		195
Зертханалық жұмыс № 1. Жалған кездейсоқ санның базалық генераторларын Зерттеу.....		195
Зертханалық жұмыс № 2. Жалған кездейсоқ сан генераторларының сапасын Тексеру.....		197
Зертханалық жұмыс № 3. Берілген таралу заңы бар кездейсоқ шаманы Түрлендіру.....		198
Зертханалық жұмыс № 4. Қалыпты таралу заңы бар кездейсоқ шаманы Түрлендіру.....		202
Зертханалық жұмыс № 5. Жиі қолданатын тарату заңы бар кездейсоқ шаманы түрлендіру.....		203
Зертханалық жұмыс № 6. Монте-Карло әдісімен модельдеу.....		203
Зертханалық жұмыс № 7. Кездейсоқ кезуді модельдеу.....		208
Зертханалық жұмыс № 8. Жаппай қызмет көрсететін жүйені Модельдеу.....		210
Зертханалық жұмыс № 9. Тәжірибені тактикалық жоспарлау.....		214
Зертханалық жұмыс № 10. Piłgrim модельдеу құрал тәсілінің көмегімен жаппай өызмет көрсету жүйелерін модельдеу.....		214
Әдебиеттер тізімі.....		215

Оқу баспасы

**Овечкин Геннадий Владимирович,
Овечкин Павел Владимирович**

Компьютерлік модельдеу

Оқулық

Редактор Л. В. Толочкова, Ә.Шапауов
Компьютерлік беттеу: Л. М. Беляева
Түзетуші Л. В. Гаврилина

Шығ. № 101115958. Қол коюға жіберілді 29.09.2014. Пішімі 60 x 90/16.
Гарнитурасы «Балтика». Қағаз офсетті. Басу офсетті. Шар. бас. Парағы 14,0.
Таралымы 2500 дана. Тапсырыс №

«Академия» баспа орталығы». ЖШС www.academia-moscow.ru 129085, Мәскеу,
Мир д-лы, 101В, б. 1.
Тел./факс: (495) 648-0507, 616-00-29.
29.04.2014 № РОСС RU. АЕ51. Н16592 санитарлы-эпидемиологиялық қорытынды.
Баспаның электронды тасығышынан басылған.

«Тверь полиграфиялық комбинаты» АОҚ, 170024, Тверь қ., Ленин д-лы, 5.
Телефон: (4822) 44-52-03, 44-50-34. Телефон/факс: (4822) 44-42-15.
Homepage—www.tverpk.ru Электронды пошта(E-mail) —sales@tverpk.ru