

Заключение

Анализ текущего состояния информационной безопасности выявил, что основными источниками рисков на уровне банковского отделения являются фрагментарность организационных решений, неоднородность программно-аппаратной инфраструктуры и влияние человеческого фактора. Формальное выполнение требований безопасности без их адаптации к реальным условиям функционирования отделения снижает эффективность защитных мер и не позволяет своевременно реагировать на изменение угрозой обстановки. Сформированная стратегия повышения уровня защищённости ориентирована на согласование мер безопасности с бизнес-процессами банковского отделения, обеспечение адаптивности системы защиты и её интеграцию в процессы управления и контроля. Такой подход позволяет не только снизить вероятность реализации актуальных угроз, но и повысить устойчивость информационной инфраструктуры в долгосрочной перспективе. Модернизация средств защиты, развитие системы управления доступом, повышение уровня осведомлённости персонала и регулярная оценка эффективности принятых решений позволяют сформировать целостную и управляемую систему информационной безопасности на уровне банковского отделения. Таким образом, стратегия повышения информационной безопасности банковского отделения должна рассматриваться как непрерывный процесс, направленный на управление рисками, адаптацию к изменяющимся условиям и поддержание доверия клиентов.

Список литературы

1. Столлингс У. Криптография и защита сетей: принципы и практика. М.: Вильямс, 2020.
2. Шнайер Б. Прикладная криптография. М.: Диалектика, 2019.
3. Таненбаум Э., Уэзеролл Д. Компьютерные сети. СПб.: Питер, 2021.
4. Pfleeger C., Pfleeger S., Margulies J. Security in Computing. Pearson, 2020.
5. ISO/IEC 27001:2022. Information Security Management Systems – Requirements.

МОДЕЛЬ ЯСТРЕБ-ГОЛУБЬ ДЛЯ АНАЛИЗА АГРЕССИВНОГО И ДРУЖЕЛЮБНОГО ВЕБ-ИНТЕРФЕЙСА

Попов Т. И., Покуса Т. В.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: tamilapokusa@mail.ru*

Научный руководитель: Покуса Т. В.

Введение

Современный интернет имеет крайне широкий спектр выбора для пользователей среди сайтов предоставляющие услуги. Регулярно случа-

ются инциденты, когда интернет-пользователи могут столкнуться с мошенническим сайтом, или как минимум – с сайтом использующие человеческие качества пользователя для того, чтобы повысить конверсию своей площадки. Такой подход зачастую агрессивен с точки зрения дизайна и маркетинга, с другой стороны существуют сайты с повышенной прозрачностью и вызовом доверия у пользователей, но они парадоксально имеют куда меньший охват пользователей, чем сайты с агрессивным подходом. С точки зрения теории игр, эти две категории сайтов можно поделить на две группы: «Ястребы» и «Голуби». Согласно математической модели «Ястребы и голуби», её смысл состоит в описывании конкурентных отношений в некоторой популяции животных и выработки оптимальной эволюционной стратегии. Данная модель будет адаптирована для экосистем рынка веб-сайтов: «Ястреб» – сайты использующие методы скрытого воздействия в дизайне и громкую рекламу, «Голуби» – сайты менее заметные, чем «ястребы» из-за меньшей рекламной политики и имеющие большую степень честности по отношению к покупателям. Используя вышеуказанную модель из теории игр, в данной научной работе будет исследоваться потенциально наиболее выгодная модель для рынка и при каких условиях рынок может достичь равновесия в конкуренции «голубей» и «ястребов». Практическая значимость состоит в том, что формализация конфликта даёт чёткие критерии, по которым пользователи распознают манипулятивные сайты. Модель позволяет владельцам площадок и аналитикам проверить, насколько выгодна та или иная стратегия при сложившихся условиях рынка, а также определить долю «ястребов» и «голубей», при которой рынок переходит в устойчивое состояние.

Цель исследования – нахождение точки равновесия в интернет-рынке для сайтов из групп «ястребов» и «голубей». Вычисление наиболее оптимальной стратегии поведения для веб-сайтов. Найти способ распознавать сайты использующие манипуляции.

Материалы и методы исследования

В работе используется метод из теории игр «Ястреб и голубь». Джон Мейнард Смит описал ситуацию, в которой животные выбирают: уступить или драться. Это вариант игры «Струсил – проиграл». В споре за добычу оба соперника обычно проявляют агрессию и стремятся ранить друг друга. Непосредственно перед схваткой каждый решает: отступить, оставив добычу, но оставившись живым (голубь), или продолжать бой до конца, рискуя погибнуть (ястреб). Предположим, в популяции, где все ведут себя как голуби, появляется небольшая партия ястребов. Сначала доля ястребов увеличивает-

ся, потому что их поведение приносит выгоду: при встрече голубя и ястреба побеждает последний. По мере того как ястребов становится больше, сталкиваются они чаще, а потому травмы между ними учащаются. В конце концов устанавливается устойчивое соотношение между ястребами и голубями. В природе наблюдается именно такой баланс. С учётом сказанного, была создана платёжная матрица Смитом. Платежи были бы определены следующим принципом: достижение цели (V) оценивается в 10 очков, увечья (W) приносят – 20 очков. В схватке ястребов между собой, победа и поражение чередуются и каждый получает – 5 очков, в схватке ястреба и голубя, ястреб всегда побеждает и получает 10 очков, а голубь отстывает (L) и получает 0 очков. В схватке голубей пострадавших нет, но голуби тратят время (E), оценка ситуации в – 3 очка. В голубиной схватке победитель получает $10 - 3 = 7$ очков, проигравший – 3 очка, в среднем каждый получает по 2 очка [1].

	Ястребы	Голуби
Ястребы	(-5; -5)	(10; 0)
Голуби	(0; 10)	(2; 2)

Существуют формулы для вычисления платёжной матрицы:

$$S(\Gamma, \Gamma) = ((V - L) / 2) - E;$$

$$S(\mathcal{A}, \mathcal{A}) = (V - W) / 2;$$

$$S(\Gamma, \mathcal{A}) = -L;$$

$$S(\mathcal{A}, \Gamma) = V.$$

Также будут в научной работе будут использоваться формулы для вычисления:

среднего выигрыша ястребов

$$S_H(z) = z \times S(\mathcal{A}, \mathcal{A}) + (1 - z) \times S(\mathcal{A}, \Gamma),$$

среднего выигрыша голубей

$$S_D(z) = z \times S(\Gamma, \mathcal{A}) + (1 - z) \times S(\Gamma, \Gamma),$$

нахождения доли ястребов

$$z = \frac{V + L + 2E}{W - L + 2E}.$$

Метод «Ястреб и голубь» требует адаптации под цель научной работы – анализ веб-сайтов, поэтому для поиска переменных нельзя в точности соблюдать классическому методу. 1) Переменная V – конверсия сайта, находится путём подсчёта решений через таблицу в дизайне нацеленных на повышение конверсии сайта, каждая реализация добавляет сайту 1 балл, отсутствие решения даёт 0 баллов. 2) Переменная W – методы скрытого воздействия, находится аналогичным образом, подсчитываются элементы интерфейса, что используют манипулятивные техники над пользователями для дости-

жения своих целей. Дополнительно, в таблице чёрных паттернов объясняется опасность того или иного решения с ссылкой на книгу о манипулятивных приёмах в дизайне – «Evil by design» за авторством Криса Ноддера [2]. 3) Переменная E – потраченное время пользователя для принятия решения, является когнитивной нагрузкой на покупателя, чем больше времени уходит у человека, чтобы считать элементы сайта, тем выше вероятность, что он его покинет. Когнитивная нагрузка высчитывается по закону Хика. Закон Хика, также называемый законом Хика–Хаймана, назван в честь британского психолога Уильяма Эдмунда Хика и американского психолога Рэя Хаймана. Он описывает, как растёт время, требующееся человеку для выбора, при добавлении новых альтернатив: чем больше появляется возможных вариантов, тем дольше приходится ждать, пока человек примет решение, причём это увеличение следует логарифмическому закону. Закон Хика имеет следующую формулу

$$L = b \times \log_2(n + 1),$$

где b – константа, которая может быть определена эмпирически путем подгонки линии к измеренным данным. Добавление 1 к n учитывает «неопределенность относительно того, отвечать или нет, а также относительно того, какой ответ дать» [3]. На место константы b , будет ставиться конкретное значение из исследования «Making Up Your Mind After a 100-Ms Exposure to a Face» за авторством Дженина Уиллиса и Александра Тодорова [4]. В исследовании доказывалось научным методом, что людям хватает 100 миллисекунд, чтобы сформировать мнение о человеке, а увеличение время просмотра на лицо, в большей степени, лишь закрепляет мнение человека [4]. Таким образом, 100 миллисекунд равняются 0.1 секунде и данное значение будет использоваться как значения b в формуле. Вместо значения n подставляется общая сумма интерактивных элементов на сайте, что подсчитываются визуально. Стоит учесть, что исследования Хика не является полностью объективным определением влияния сложности восприятия объекта на реакцию человека. Закон Хика, трактуют время реакции (Reaction Time – RT) как прямое отражение длительности когнитивных процессов, современные исследования ставят эту трактовку под сомнение. Работа Wong A. L. (2017) показывает, что RT может быть в значительной степени подвержен опытозависимому смещению, функционируя как независимый параметр движения, устанавливаемый по привычке [5]. Это означает, что увеличение времени реакции в условиях множественного выбора может отражать не только рост вычислительной сложности, но и формирование адаптивной двигательной стратегии в ответ на неопреде-

ленность. Таким образом, интерпретация данных RT, в том числе в контексте закона Хика, требует учёта истории двигательного опыта испытуемого и допущения, что время реакции несёт в себе как вычислительную, так и привычную компоненту [5]. Тем не менее, закон Хика всюду используется современными дизайнерами и маркетологами для оценки сложности восприятия сайтов, по данной причине, нахождение переменной E будет проводиться данным путём. 4) Значение L из метода «ястреба и голубя» не находится, и всегда равняется 0. Само исследование происходит следующим образом.

Для реализации методологии анализа дизайн-стратегий формируются две группы сайтов: группа «голубь» и группа «ястреб», по два сайта в каждой. Для повышения репрезентативности сравнения каждый исследуемый сайт будет парно сопоставляться с прямым или косвенным конкурентом из противоположной группы, действующим на одном рыночном сегменте. Это позволит минимизировать влияние отраслевой специфики на результаты оценки переменных V, W и E. Экспертная оценка каждого сайта по всем критериям будет проводиться независимо, после чего полученные баллы усредняются внутри каждой группы. Это даёт усреднённые групповые значения $V_{\text{группа}}, W_{\text{группа}}, E_{\text{группа}}$, отражающие типичные характеристики стратегий «голубь» и «ястреб». Для построения единой платёжной матрицы, описывающей взаимодействие в общей популяции сайтов, необходимо рассчитать общие средние значения переменных V, W и E по всей выборке. Эти финальные значения $V_{\text{общее}}, W_{\text{общее}}, E_{\text{общее}}$, наряду с константой $L = 0$, и подставляются в формулы Смита для расчёта выигрышей в клетках матрицы:

$$S(\Gamma, \Gamma) = ((V_{\text{общ}} - L) / 2) - E_{\text{общ}};$$

$$S(\mathcal{A}, \mathcal{A}) = (V_{\text{общ}} - W_{\text{общ}}) / 2;$$

$$S(\Gamma, \mathcal{A}) = -L;$$

$$S(\mathcal{A}, \Gamma) = V_{\text{общ}};$$

Именно на основе этой общерыночной матрицы затем вычисляется средний выигрыш ястребов $S_H(z)$ и голубей $S_D(z)$ в зависимости от доли ястребов z, а также находится точка равновесия z^* , при которой выгоды от обеих стратегий сравниваются и рынок достигает устойчивого состояния.

Результаты исследования и их обсуждение

Основа исследования собрана из четырёх веб-сайтов. Два из них – онлайн-кинотеатры, два других – сервисы пополнения счёта в игровой платформе Steam. В каждой паре выбраны конкурирующие ресурсы с противоположными подходами к пользователю. Среди кинотеатров сайт Ivi показывает агрессивную стратегию «Ястреба», а «Кинопоиск» – честную стратегию «Голубя». Среди платёжных сервисов поведение «Ястреба» иллюстрирует Kupikod, а «Голубя» – PayBerry. Первый качественный этап оценил переменные модели. Для пары Ivi и «Кинопоиск» в таблице 1 собраны скрытые манипулятивные приёмы, распределённые по типологии Криса Ноддера в книге «Evil by Design» [2]. Эти данные дали значение переменной W – потенциал негатива. В таблице 2 для тех же сайтов перечислены дизайн-решения, повышающие конверсию; по ним вычислена переменная V – ценность ресурса. Аналогичный разбор проведён для пары Kupikod и PayBerry. Заключительный этап формализует собранные данные в модели «Ястреб и Голубь»: вычисляются средние значения, строится платёжная матрица и находится точка равновесия. «Ястреба и голубя».

Таблица 1

Анализ веб-сайтов «Ivi» и «Кинопоиск» в области онлайн-кинотеатров

№	Скрытая манипуляция	Кинопоиск	Ivi
1	Принудительная непрерывность и сложная отмена	1	1
2	Скрытая информация	0	1
3	Принудительный конструктор	1	1
4	Имитация срочности	0	1
5	Имитация редкости	0	1
6	Скрытое подтверждение	0.	1
7	Социальное доказательство	0	1
8	Привязка к аккаунту	0	1
9	Навязчивые уведомления	1	1
10	Плата за удобство	1	1

Кинопоиск: $W = 4$;

Ivi: $W = 10$.

№	Дизайн-решение, повышающее конверсию	Кинопоиск	Ivi
1	Бесплатный просмотр первой серии	1	1
2	Автовоспроизведение следующей серии	1	1
3	Массовая реклама	1	1
4	Эмоциональные постеры	1	1
5	Чёткое разделение бесплатного и платного	1	0
6	Пакетные предложения (бандлы)	1	0
7	Агрессивные уведомления о премьерах	0	1
8	Многоуровневая система подписок	0	1
9	Персонализированные рекомендации	1	1
10	Единый аккаунт с экосистемой	1	0

Кинопоиск: $V = 8$;

Ivi: $V = 7$.

Кинопоиск: $E = 0.1 \times \log_2(8+1) = 0.316$ секунды.

Ivi: $E = 0.1 \times \log_2(11+1) = 0.358$ секунды.

Таблица 2

Анализ сайтов «Kupikod» и «PayBerry» в области пополнения аккаунта для игрового магазина «Steam»

№	Скрытая манипуляция	Kupikod	PayBerry
1	Скрытые дополнительные платежи	1	0
2	Искусственное создание ажиотажа	1	0
3	Затруднённая процедура отказа от услуг	1	0
4	Автоматическое подключение платных опций	1	0
5	Использование внутренней виртуальной валюты	1	0
6	Установка завышенного ценового ориентира	1	0
7	Привлечение авторитетных мнений	1	0
8	Система поощрений за совершение действий	1	1
9	Иллюзия выгоды от увеличения суммы	1	0
10	Неисполнение обязательств после оплаты	1	0

Kupikod: $W = 10$;

PayBerry: $W = 1$.

№	Дизайн-решения повышающие конверсию	Kupikod	Payberry
1	Активное саморекламирование	1	0
2	Возможность совершить транзакцию с главной страницы	1	0
3	Реклама самых выгодных условий	1	0
4	Подключение к подписке после первой покупки	1	0
5	Визуализация этапов заполнения данных необходимых для транзакции	1	0
6	Подробные инструкции как совершить транзакцию	1	1
7	Снижении комиссии при повышении суммы совершаемой транзакции	1	0
8	Чистый и понятный дизайн	0	1
9	Отображение снимаемой суммы с учётом комиссии	1	1
10	Получение чека	0	1

Kupikod: $V = 8$;

PayBerry: $V = 4$.

Kupikod: $E = 0.1 \times \log_2(6+1) = 0.281$ секунды.

PayBerry: $E = 0.1 \times \log_2(20+1) = 0.439$ секунды.

3. Вычисление результатов по методу «Ястреба и голубя» используя полученные данные.

Группа «голубей»:

Кинопоиск ($V = 8; W = 4; E = 0.316$),

PayBerry ($V = 4; W = 1; E = 0.2807$).

Группа «ястребов»:

Ivi ($V = 7; W = 10; E = 0.358$),

Курикод ($V = 8; W = 10; E = 0.439$).

Средние по группе «Голубь»:

$$V = \frac{(8+4)}{2} = 6.0$$

$$W = \frac{(4+1)}{2} = 2.5$$

$$E = \frac{(0.281+0.316)}{2} = 0.2985$$

Средние по группе «Ястреб»:

$$V = \frac{(7+8)}{2} = 7.5$$

$$W = \frac{(10+10)}{2} = 10.0$$

$$E = \frac{(0.439+0.358)}{2} = 0.3985$$

Общерыночные средние (ключевые для матрицы):

$$V = \frac{(6.0+7.5)}{2} = 6.75$$

$$W = \frac{(2.5+10.0)}{2} = 6.25$$

$$E = \frac{(0.2985+0.3985)}{2} = 0.3485$$

Перевод общих средних значений в баллы модели:

$$V = \frac{6.75}{10} \times 10 = 6.75;$$

$$W = \left| -\frac{6.25}{10} \right| \times 20 = 12.5;$$

$$E = -\left(\frac{0.3485}{0.439} \right) \times 3 = -2.38;$$

$$L = 0;$$

Построение платёжной матрицы:

$$S(G, G) = \frac{6.75}{2} - (-2.38) = 5.755;$$

$$S(Y, Y) = \frac{6.75 - 12.5}{2} = -2.875;$$

$$S(G, Y) = 0;$$

$$S(Y, G) = 6.75;$$

	Ястребы	Голуби
Ястребы	(-2.875; -2.875)	(6.75; 0)
Голуби	(0; 6.75)	(5.755; 5.755)

Нахождение равновесия доли ястребов (Z):

$$z = \frac{6.75 + 0 + 2 * (-2.38)}{12.5 - 0 + 2 * (-2.38)} = 0.257$$

$$S_Y(z) = 0.257 * (-2.875) + (1 - 0.257) * 6.75 = 4.276$$

$$S_G(z) = 0.257 * 0 + (1 - 0.257) * 5.755 = 4.275 .$$

Заключение

Исследование адаптировало стандартную игровую модель «ястреб и голубь» для анализа конкуренции между веб-сайтами. В работе выделили четыре переменные – V, W, E, L – и показали, как оценить каждую: дизайн-эксперт задаёт значения, а когнитивную нагрузку вычисляются по закону Хикса. Методику проверили на четырёх сайтах, сгруппированных в две пары-конкурента: одна пара из онлайн-кинотеатров, вторая из сервисов цифровых платежей. Эксперты зафиксировали чёткое разделение стратегий: «ястребы» скрыто манипулируют пользователем через тёмные паттерны, а «голуби» делают интерфейс прозрачным и убирают лишние шаги. Анализ получившихся среднерыночных параметров ($V = 6.75; W = 12.5; E = -2.38, L = 0$) и построенная на их основе платёжная матрица показали эволюционно-стабильное состояние рынка. Модель выдала, что равновесие наступает тогда, когда доля площадок, выбравших агрессивную стратегию «ястреб», равняется $z^* \approx 25.7\%$, а оставшиеся 74.3% придерживаются честной стратегии «голубь». В этой точке средний выигрыш каждой стратегии одинаков и равен примерно 4.28 условных единиц. Результат важен как для теории, так и для практики. В теории он показывает: даже при заметном конверсионном преимуществе агрессивных сайтов ($V_{\text{ястреб}}$ выше $V_{\text{голубь}}$), их рост тормозится большими издержками взаимной конкуренции. Когда встречаются два «ястреба», выигрыш становится отрицательным (-2.875), поэтому у «шолубей» формируется устойчивая экологическая ниша. В практическом плане рассчитанная доля агрессоров (25.7%) меньше той, что зафиксирована в выборке (50%), значит при сохранении текущих условий рынок может сместиться к более честным практикам. Модель формализует конфликт интересов и даёт численную базу для прогноза дизайн-стратегий: в долгосрочной перспективе стратегия

«голубь» не только этически предпочтительна, но и эволюционно устойчива, занимая большую часть равновесия.

Список литературы

1. Деулофеев Х. Теория игр. Дилемма заключённого и доминантные стратегии. Москва: Ленанд, 2014. 272 с.
2. Nodder C. *Evil by Design: Interaction Design to Lead Us into Temptation*. Indianapolis: John Wiley & Sons, 2013. 258 p.
3. Hick W. E. On the rate of gain of information // *Quarterly Journal of Experimental Psychology*. 1952. Vol. 4. No. 1. P. 11–26.
4. Willis J., Todorov A. First Impressions: Making Up Your Mind After a 100-ms Exposure to a Face // *Psychological Science*. 2006. Vol. 17. No. 7. P. 592–598.
5. Wong A. L., Goldsmith J., Forrence A. D., Haith A. M., Krakauer J. W. Reaction times can reflect habits rather than computations // *eLife*. 2017. Vol. 6. Article e28075. P. 1–18. DOI: 10.7554/eLife.28075.

СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ JAVA И C++ ПРИ СОРТИРОВКЕ ЦЕЛЫХ ЧИСЕЛ И ЗАПИСИ/ЧТЕНИИ ФАЙЛОВ

Романенко И. В.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: ryukjwfr@gmail.com*

Научный руководитель: Ступницкий В. С.

Введение

Выбор языка программирования оказывает существенное влияние на производительность, надёжность и сопровождаемость программного обеспечения. Особенно важен этот выбор при разработке приложений, требующих эффективной обработки данных и выполнении базовых операций, таких как сортировка и работа с файлами. Среди множества языков особое внимание заслуживают C++ и Java, два объектно-ориентированных языка, широко применяемых в промышленной разработке, но реализованных на разных уровнях системной абстракции. C++ компилируемый язык, близкий к аппаратному уровню, предоставляющий разработчику полный контроль над ресурсами системы. Однако эта гибкость сопряжена с повышенной сложностью: ошибки в управлении памятью (например, утечки или двойное освобождение) могут привести к нестабильной работе. Этот недостаток можно смягчить за счёт строгого соблюдения правил RAII (Resource Acquisition Is Initialization) и использования современных средств языка, таких как автоматические локальные переменные и контейнеры стандартной библиотеки (например, `std::vector`), которые управляют памятью самостоятельно. Java, напротив, выполняется в среде виртуальной машины и обеспечивает автоматическое управление памятью с помощью сборщика мусора. Это повышает надёжность и упрощает разработку, но вносит накладные расходы, что может сни-

зить производительность в вычислительных интенсивных задачах. Этот недостаток так же можно частично компенсировать настройкой параметров JVM и применением пулов памяти или off-heap решений. Эти особенности делают сравнение языков актуальным: несмотря на схожесть в синтаксисе и парадигмах, они по-разному влияют на эффективность выполнения одних и тех же операций. Ранние исследования (например, Prechelt, 1999; Nikishkov et al., 2003) отмечали значительное отставание Java по скорости и потреблению памяти. Однако современные версии компиляторов и виртуальных машин существенно улучшили ситуацию, что требует актуального повторного сравнения в контексте типовых задач.

Цель исследования – провести сравнительный анализ производительности языков программирования C++ и Java при выполнении базовых операций, таких как сортировка целых чисел, чтение и запись файлов, с целью выявления их относительных преимуществ и недостатков в типичных сценариях использования.

Материалы и методы исследования

Для сравнения производительности языков C++ и Java были созданы два приложения с идентичным дизайном, осуществляющие загрузку, сортировку и сохранение массивов целых чисел. Эксперимент проводился на наборах данных разного объёма (1000, 10 000, 100 000 элементов). Время выполнения каждой операции измерялось отдельно. Для измерения времени в C++ использовалась стандартная библиотека `ctime`. Она содержит тип `clock_t` и функцию `clock`. Вместе они позволяют получить количество тактов процессора с момента запуска системы. Чтобы измерить длительность события (например, загрузки данных), вычитали начальное время из конечного. Для этого фиксировали время до и после события. Для Java использовался стандартный пакет `System`, в котором есть встроенная функция `currentTimeMillis`. Для измерения времени в миллисекундах использовалась функция `currentTimeMillis`. Время фиксировалось до и после события, затем вычислялась разница. У Java нет таких же возможностей для ручной оптимизации кода, как у C++, поскольку она компилируется собственным компилятором `javac` в байт-код, который затем выполняется на виртуальной машине Java (JVM). Этот компилятор автоматически оптимизирует код. Чтобы сделать сравнение максимально справедливым, среда разработки C++ была настроена на максимальную оптимизацию кода с использованием флага `O2` и запуск в режиме `Release`, что исключает накладные расходы на отладку и оптимизирует код настолько, насколько это возможно. Исходя из этого составим схему принципа работы системы (рис. 1).