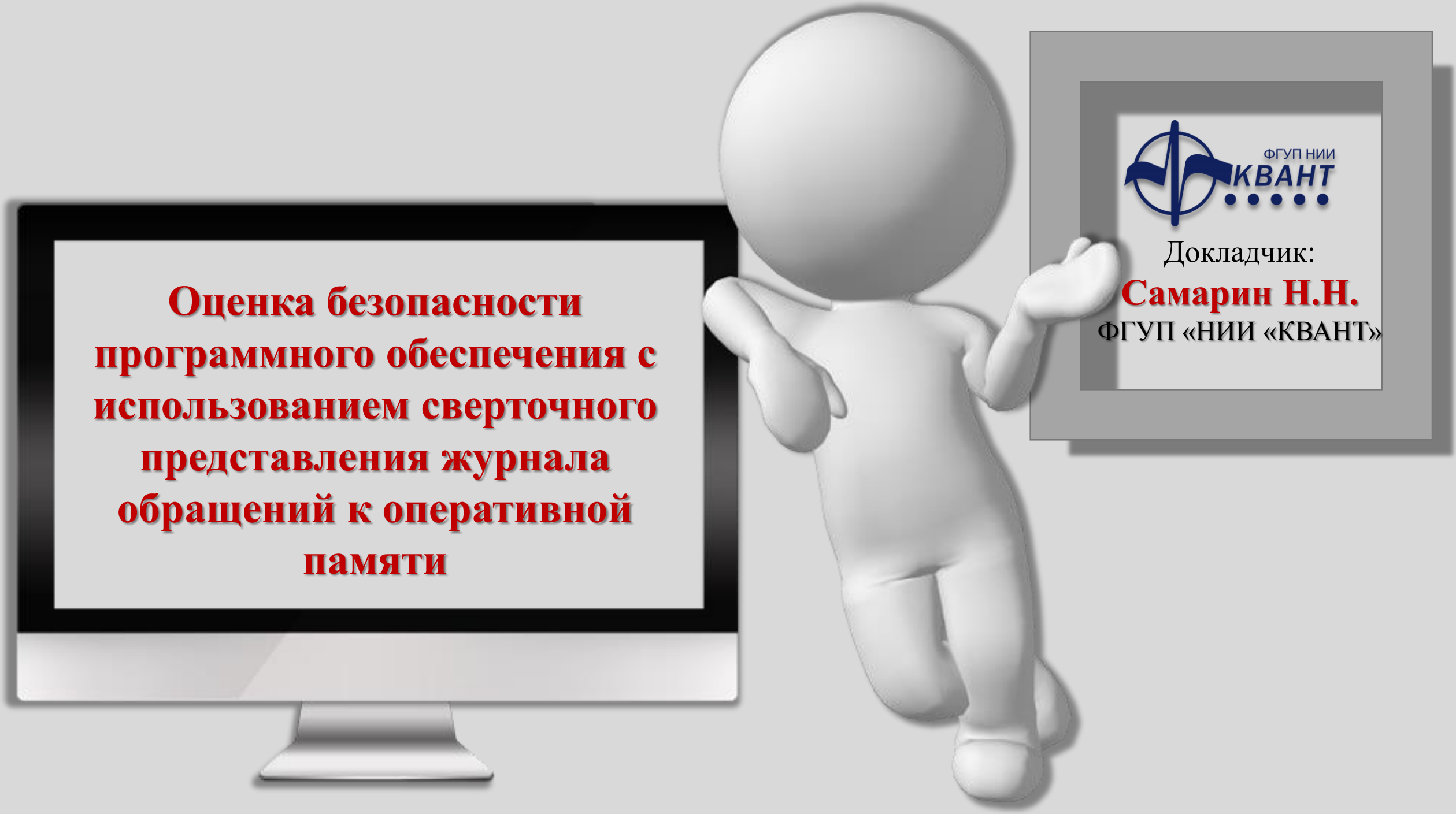
Оценка безопасности программного обеспечения с использованием сверточного представления журнала обращений к оперативной памяти



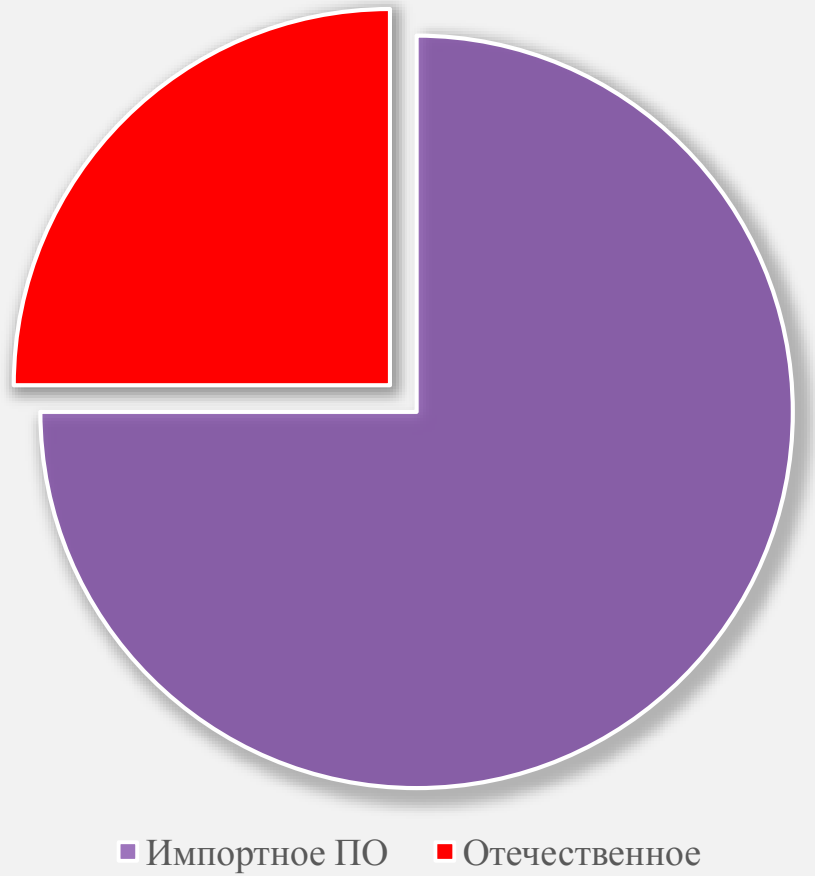
Докладчик:
Самарин Н.Н.
ФГУП «НИИ «КВАНТ»

АКТУАЛЬНОСТЬ

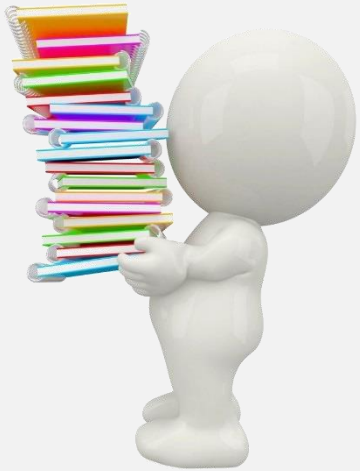
Отсутствие исходного кода приводит к усложнению:

- поиска потенциально уязвимых конструкций
- проверки на НДВ
- выявления программных закладок

Доля рынка импортного программного обеспечения на территории РФ



Классификация методов анализа ПО без исходных кодов



Методы контроля программного продукта без исходных текстов

- Предварительное дизассемблирование
  - Формирование перечня маршрутов выполнения функциональных объектов
  - Анализ сигнатур
  - Продвижение констант и копий
  - Наличие заданных конструкций в исходных текстах
  - Анализ критических маршрутов
  - Построение графа потока управления
- Контроль использования системных ресурсов (полная виртуализация)
  - Контроль и анализ областей памяти
    - Выявление действий по скрытию своего присутствия
    - Выявление действий самомодификация или изменению кода других программ
    - Выявление действий по передаче управления
    - Выявление действий по искажению результатов работы других программ
    - Выявление действий по сохранению информации
    - Выявление действий по самодублированию программ
  - Контроль команд центрального процессора
  - Контроль периферийных устройств
  - Контроль сетевых устройств

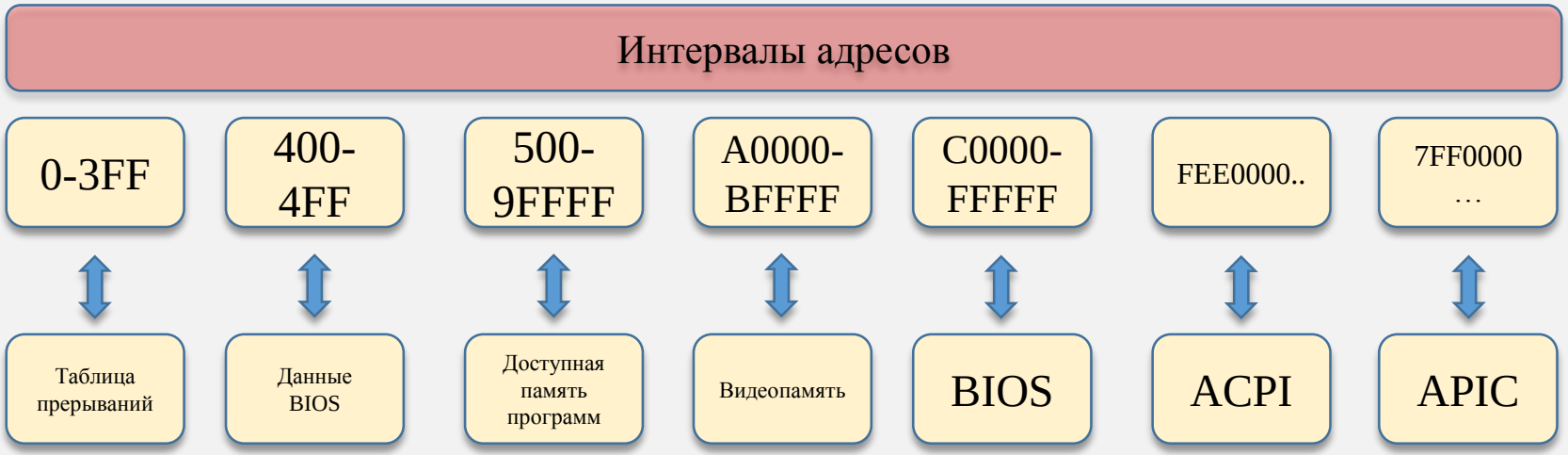
Сравнение зарубежных программных комплексов, предназначенных для оценки безопасности ПО

№ п/п	Наименование	Поддерживаемые Языки	Тип анализируемой программы		Тип анализа		Список потенциально опасных сигнатур	Определение типа уязвимостей				Наличие сертификата ФСТЭК
			С исходным кодом	Без исходного кода	Статический	Динамически		Неконтролируемая форматная строка	Переполнение буфера	Запрос исполняемых вызовов	Нулевой указатель	
1	BOON	Си	+	-	+	-	-	-	+	-	+	**нет**
2	CQual	Си	+	-	+	-	-	+	-	-	-	**нет**
3	MOPS	Си	+	-	+	+	+	+	+	+	+	**нет**
4	ITS4	Си	+	-	+	-	+	+	+	+	+	**нет**
5	RATS	Си/Си++, PHP, Perl, Python	+	-	+	-	-	+	+	+	+	**нет**
6	Flawfinder	Си/Си++	+	-	+	-	+	+	+	+	+	**нет**
7	UNO	Си	+	-	+	-	-	+	-	-	+	**нет**
8	FlexeLint (PC-Lint)	Си/Си++	+	-	+	-	-	+	+	+	+	**нет**
9	Coverity	Си/Си++, Java	+	-	+	-	-	+	+	-	+	**нет**
10	Frama-C	Си	+	-	+	-	-	+	+	+	+	**нет**
11	JavaChecker	Java	+	-	+	-	-	+	-	+	+	**нет**
12	**Qaudit**	**Си/Си++**	**+**	**-**	**+**	**-**	**-**	**+**	**+**	**+**	**+**	**нет**

Сравнение отечественных программных комплексов, предназначенных для оценки безопасности ПО

№ п/п	Наименование	Поддерживаемые Языки	Тип анализируемой программы		Тип анализа		Список потенциально опасных сигнатур	Определение типа уязвимостей				Наличие сертификата ФСТЭК
			С исходным кодом	Без исходного кода	Статический	Динамический		Неконтролируемая форматная строка	Переполнение буфера	Запрос исполняемых вызовов	Нулевой указатель	
13	АИСТ-С	Си/Си++	+	-	+	-	-	+	+	+	+	**№ 451 до 05.06.2019**
14	КСАИТ	Си/Си++	+	-	+	-	-	+	+	+	+	**нет**
15	UCA	Си/Си++, Pascal, Perl, PLM	+	-	+	-	+	+	+	+	+	**нет**
16	«АК-ВС 2»	Си/Си++, Java, Pascal, Си#, PHP, Assembler	+	-	+	+	+	+	+	+	+	**№ 3385 от 03.04.2023**
17	**Инструментальный комплекс «IRIDA 2.0»**	**Си++**	**+**	**-**	**+**	**+**	**+**	**+**	**+**	**+**	**+**	**№ 3541 до 11.05.2019**

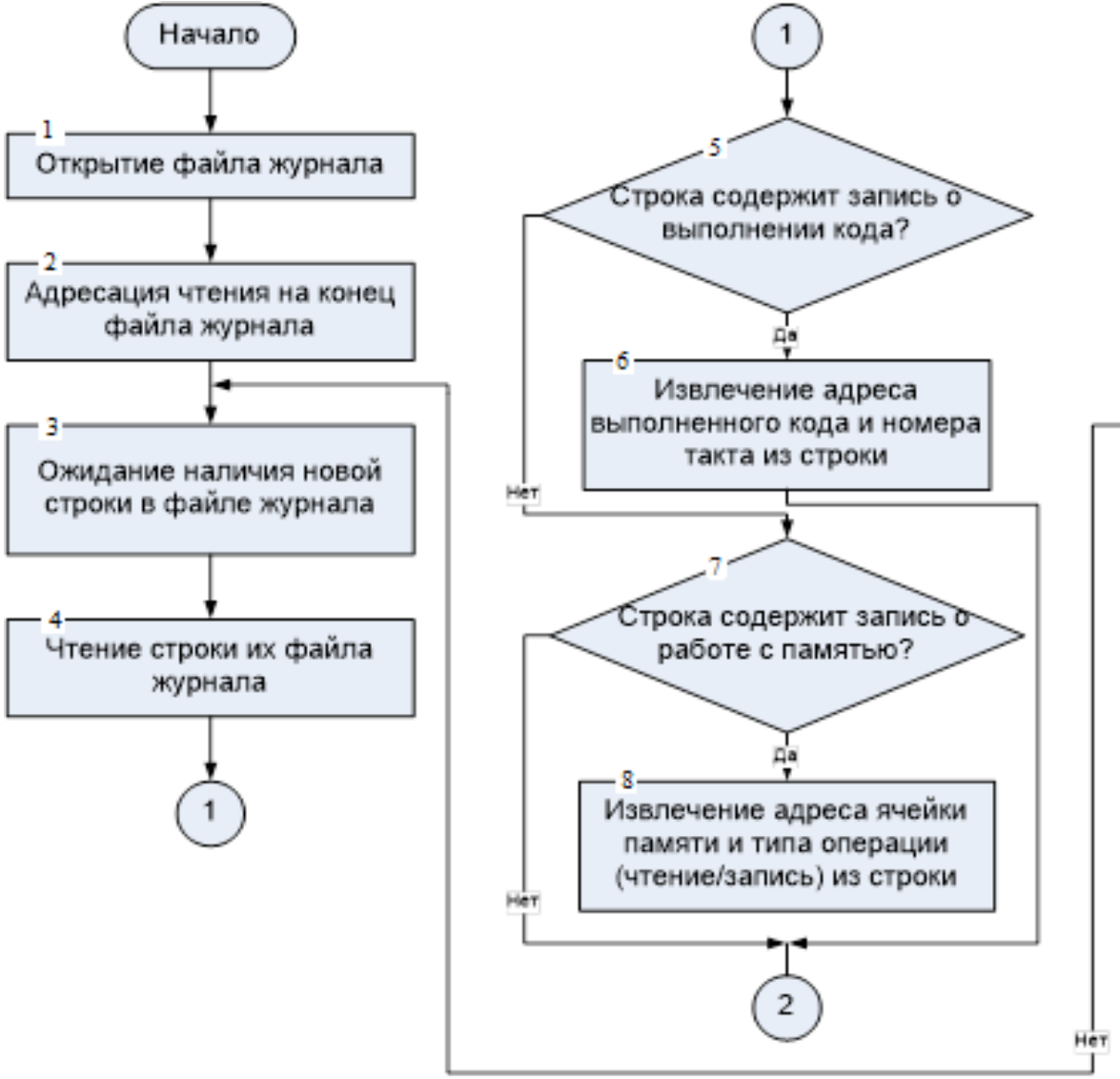
Схема распределения памяти

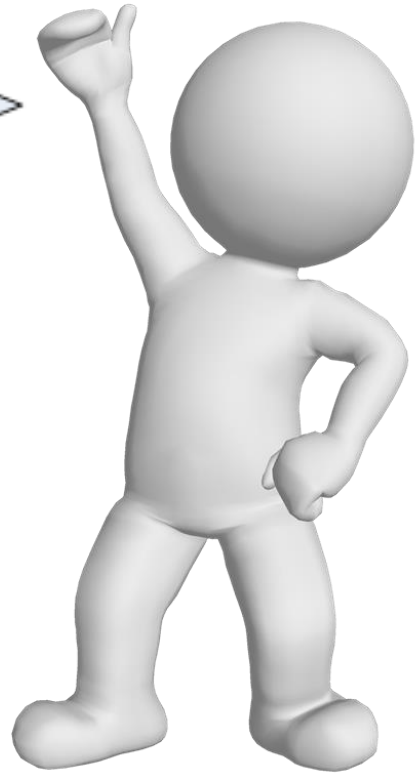


Области памяти	Типы операций процессора		
Таблица прерываний (I)	RI	WI	XI
Данные BIOS (II)	RII	WII	XII
Память программ (III)	RIII	WIII	XIII
Видеопамять (IV)	RIV	WIV	XIV
BIOS (V)	RV	WV	XV
ACPI (VI)	RVI	WVI	XVI
APIC (VII)	RVII	WVII	XVII

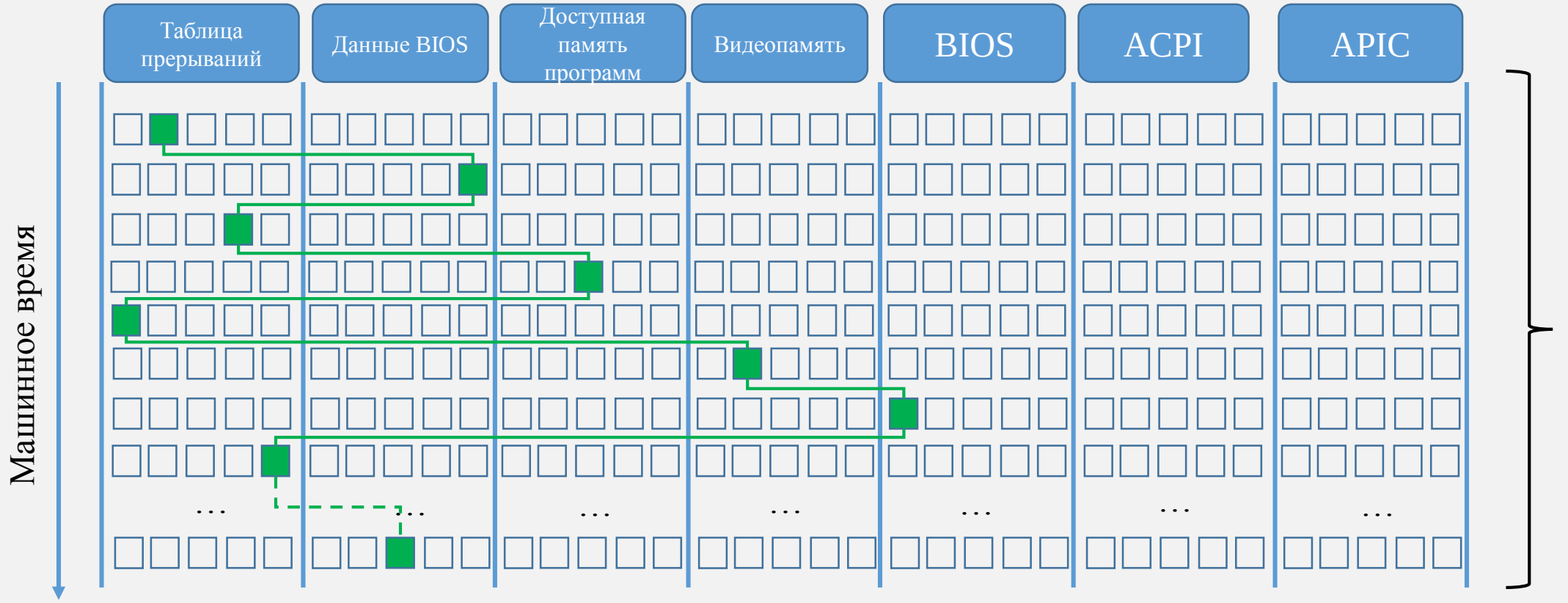
R (read)- чтение памяти
W (write)- запись памяти
X (execute)- выполнение кода, хранимого в памяти

Алгоритм контроля обращений процессора к областям памяти используемой программным продуктом





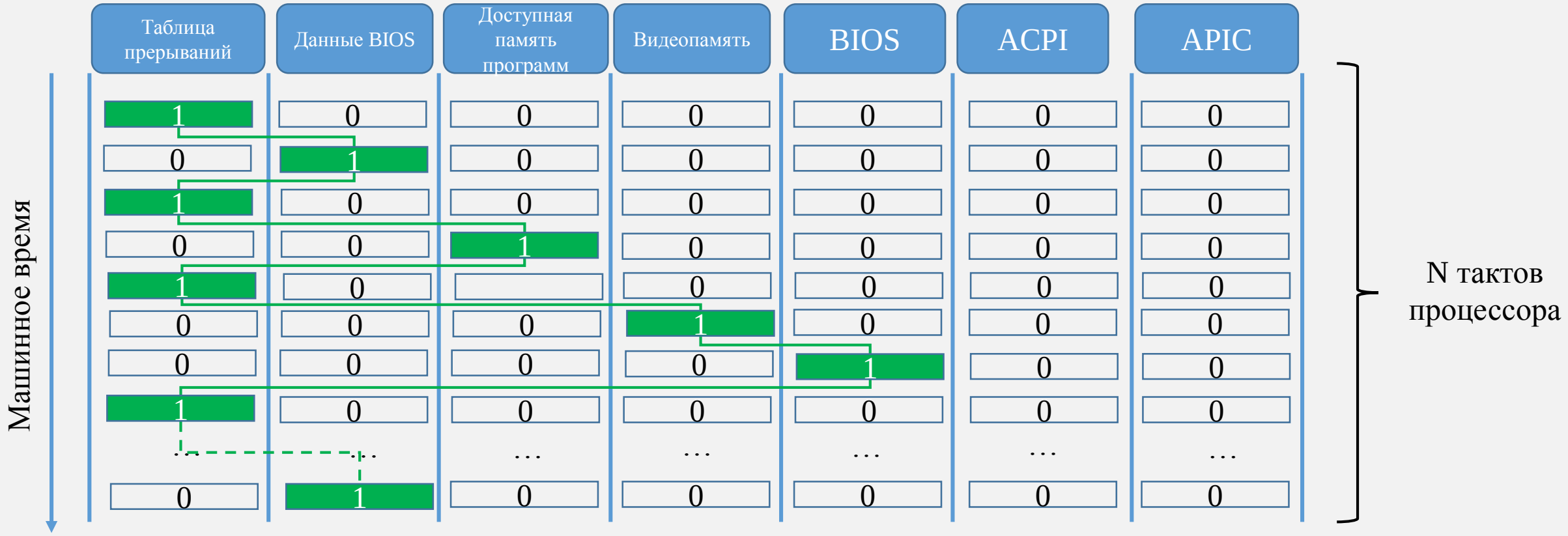
Формализованная запись обращений ПО к памяти (1/2)



Поведение ПО можно описать с помощью функции

$$F = R_I \& W_{III} \& X_I \& W_{IV} \& X_I \& R_{II} \ldots\ldots\ldots X_{II}$$

Формализованная запись обращений ПО к памяти (2/2)



Представление обращений программного обеспечения к памяти в матричном виде

7 областей памяти

N тактов процессора

Таблица прерываний	Данные BIOS	Память программ	Видеопамять	BIOS	ACPI	APIC
1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	0	0	1	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
1	0	0	0	0	0	0

Матрица размером 7xN

Сверточное представления журнала обращений программного обеспечения к памяти

№ такта	Дополнение	Таблица прерываний	Данные BIOS	Память программ	Видеопамять	BIOS	ACPI	APIC	Значение
1	0	1	0	0	0	0	0	0	@
2	0	0	1	0	0	0	0	0	-
3	0	0	0	0	1	0	0	0	BS
4	0	0	0	1	0	0	0	0	DLE
5	0	0	0	0	0	1	0	0	BS
6	0	0	1	0	0	0	0	0	-
7	0	0	0	1	0	0	0	0	DLE
8	0	0	0	0	1	0	0	0	BS
9	0	0	0	0	0	1	0	0	EOT
10	0	1	0	0	0	0	0	0	@
Итог	0	1	1	1	1	1	0	0	|

Методы машинного обучения

Машинное обучение – класс методов искусственного интеллекта, решающих новые задачи путем выявления закономерностей в существующих решениях сходных задач

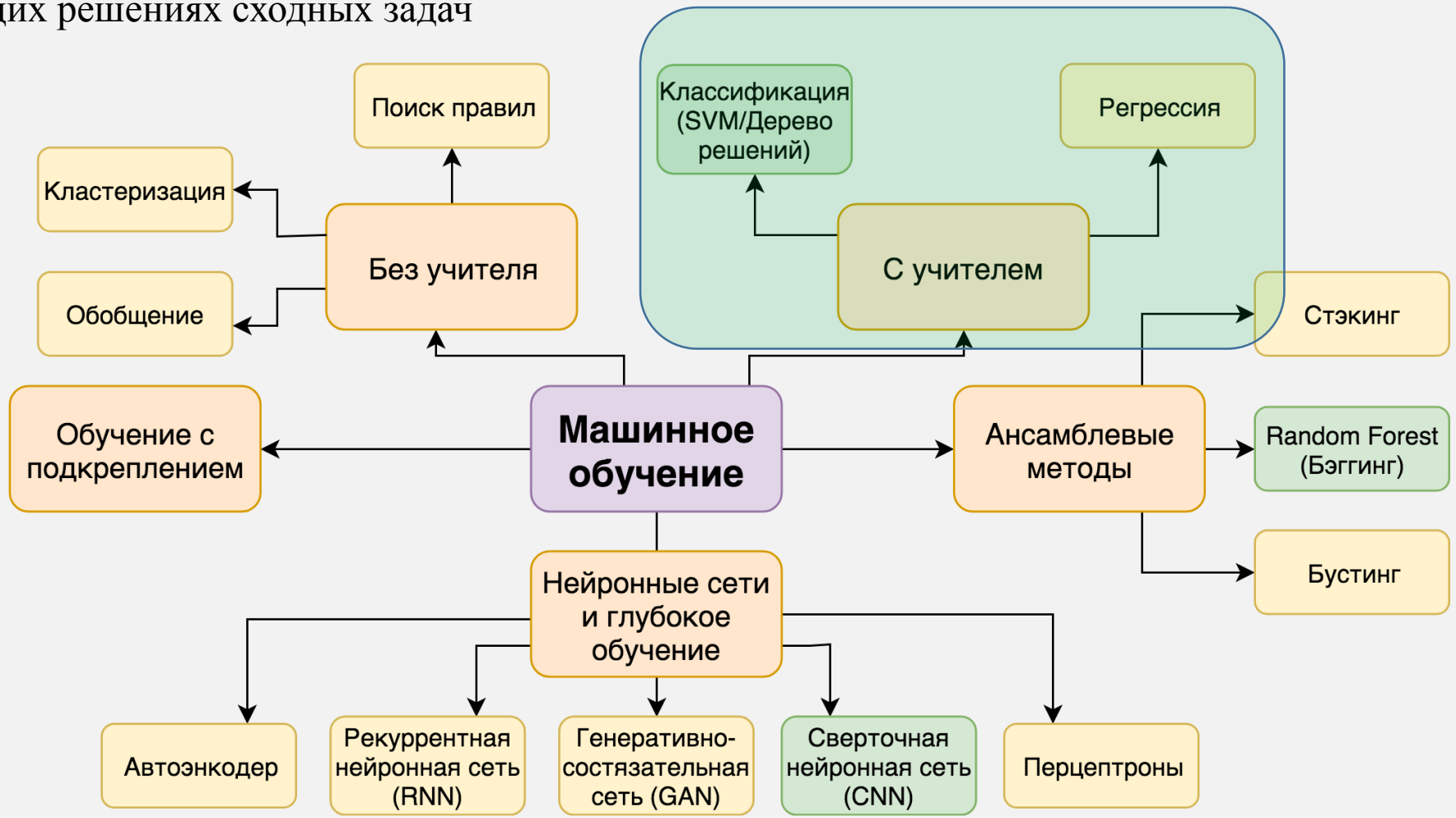


Схема работы предлагаемого комплекса оценки безопасности программного обеспечения

Настройка системы, модификация ее параметров

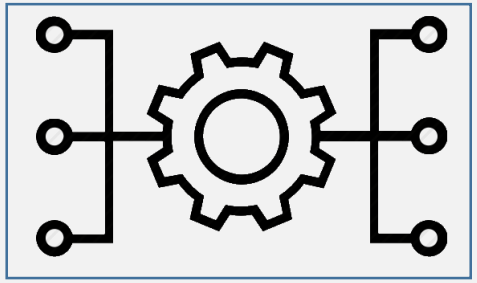
Модуль адаптации

Лог работы ПО с памятью

Машинное обучение

Анализ данных лога на предмет наличия аномалий

Принятие решения о безопасности ПО

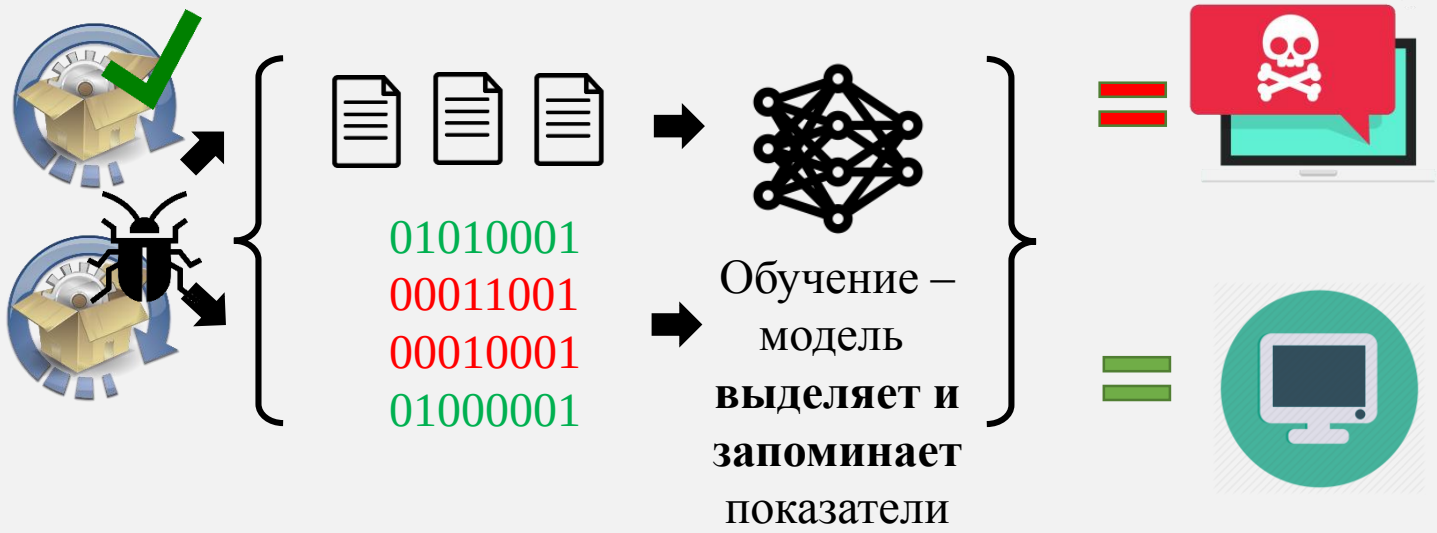


Экземпляр ПО потенциально вредоносный

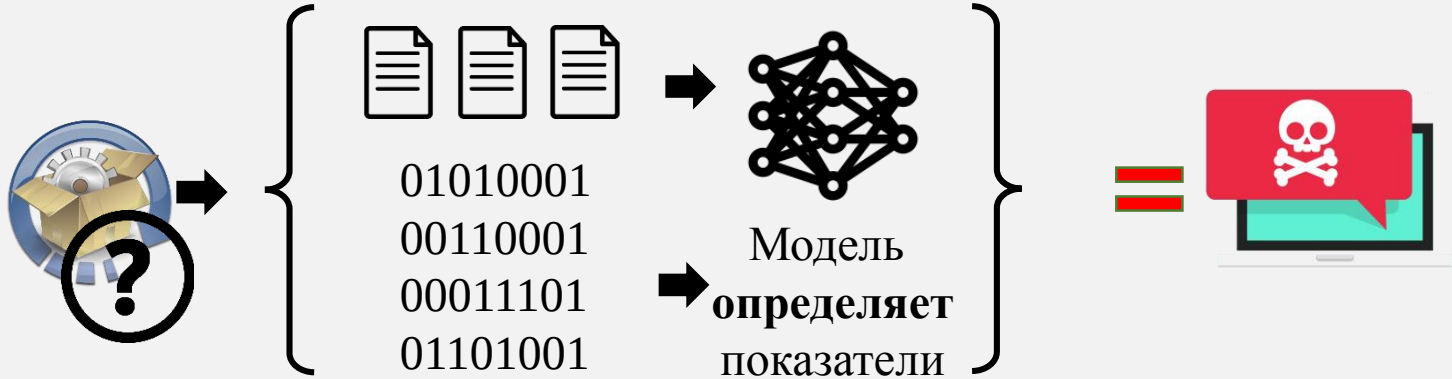
Экземпляр ПО не несет скрытых угроз

Принятие решения о безопасности программного обеспечения на основе сверточного представления журнала обращений к оперативной памяти

1 - Обучение модели на подготовленных экземплярах ПО



2 – Классификация новых Экземпляров ПО



ВЫВОДЫ

1. Проанализированы подходы к оценке безопасности ПО без исходного кода

2. Разработан алгоритм контроля обращений процессора к областям памяти используемой ПО

3. Разработан прототип для сверточного представления журнала обращений к памяти

4. Предложен подход к созданию обучающей выборки нейросети для оценке безопасности ПО


Спасибо
за внимание!