

В.В. Подбельский

# Иллюстрации к курсу лекций по дисциплине «Программирование на C#»

## 09. Часть 2

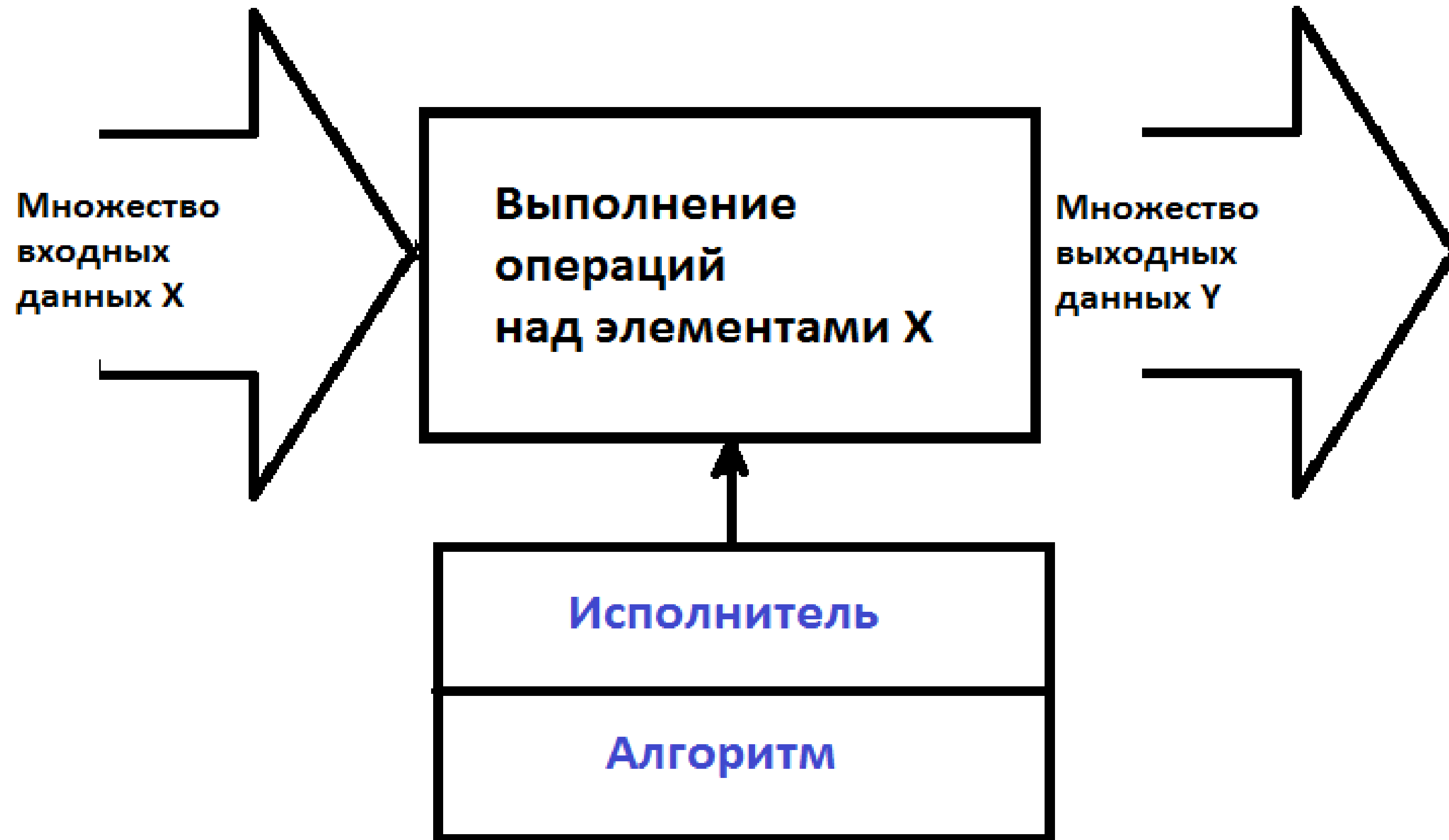
**Введение в базы данных.**

**Знакомство с Redis**

*Благодарность Андрющенко М. А. за проработку презентации*

*Визуальные правки: Сагалов Д. Д.*

# Обработка данных



# Базы данных. Системы управления базами данных

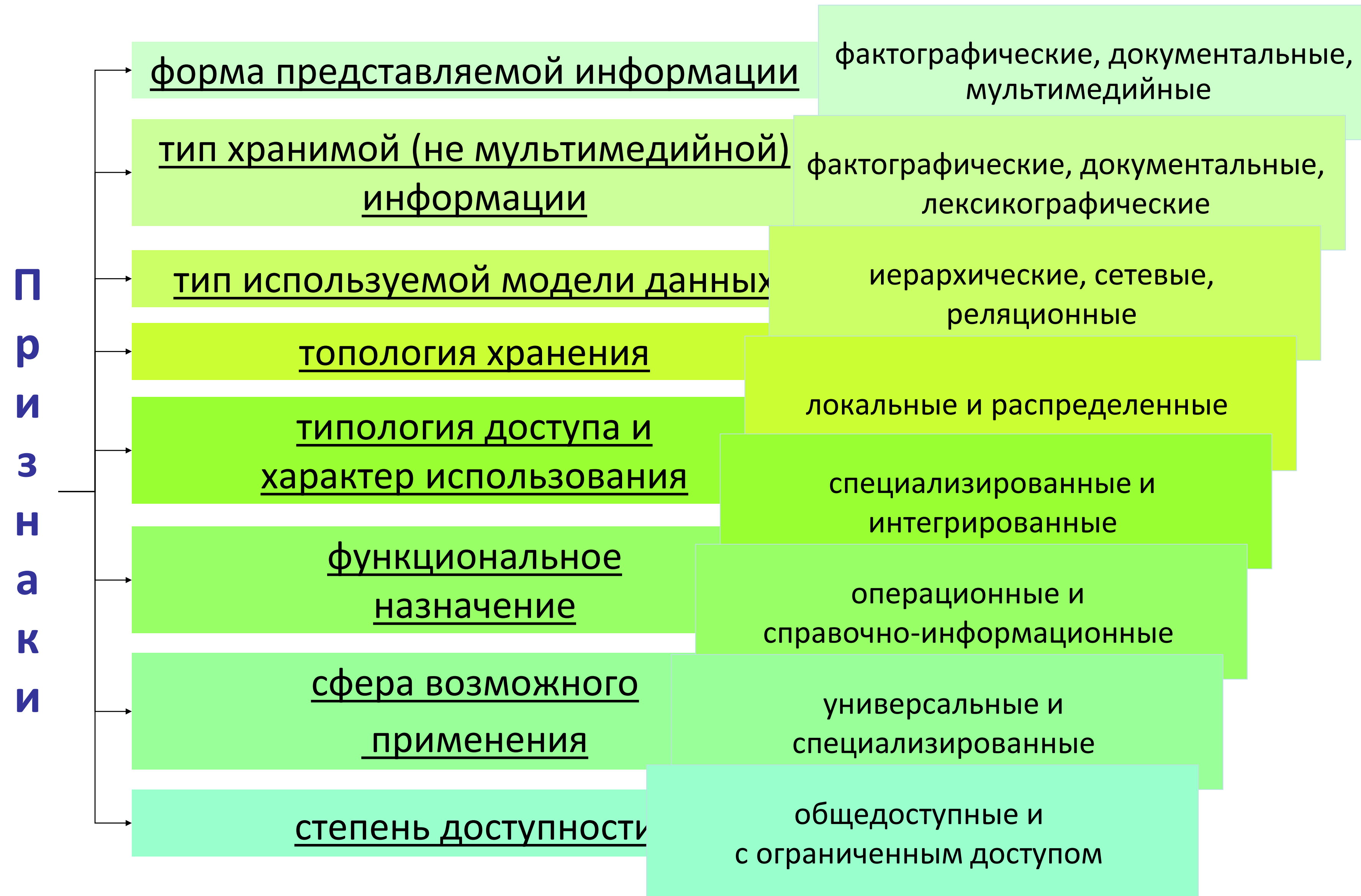
**База данных (БД)** - именованная совокупность данных, отображающая состояние объектов и их отношений в рассматриваемой предметной области.

Характерной чертой БД является **постоянство**:

- данные **постоянно** накапливаются и используются;
- структура данных, необходимая для решения тех или иных прикладных задач обычно **постоянна** и стабильна во времени.

**Система управления базами данных (СУБД)** - это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

# Классификация баз данных



# Типы современных СУБД

<https://db-engines.com/en/ranking>

381 systems in ranking, November 2021

Rank			DBMS	Database Model	Score		
Nov 2021	Oct 2021	Nov 2020			Nov 2021	Oct 2021	Nov 2020
1.	1.	1.	Oracle	Relational, Multi-model	1272.73	+2.38	-72.27
2.	2.	2.	MySQL	Relational, Multi-model	1211.52	-8.25	-30.12
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	954.29	-16.32	-83.35
4.	4.	4.	PostgreSQL	Relational, Multi-model	597.27	+10.30	+42.22
5.	5.	5.	MongoDB	Document, Multi-model	487.35	-6.21	+33.52
6.	6.	7.	Redis	Key-value, Multi-model	171.50	+0.15	+16.08
7.	7.	6.	IBM Db2	Relational, Multi-model	167.52	+1.56	+5.90
8.	8.	8.	Elasticsearch	Search engine, Multi-model	159.09	+0.84	+7.54
9.	9.	9.	SQLite	Relational	129.80	+0.43	+6.48
10.	10.	10.	Cassandra	Wide column	120.88	+1.61	+2.13

## Документоориентированные СУБД:

- MongoDB
- Amazon DynamoDB
- Microsoft Azure Cosmos DB
- Couchbase
- Firebase Realtime Database

## Другие типы СУБД:

- Graph: графовые СУБД (Neo4j)
- Spatial: пространственные СУБД (PostGis)
- Time Series: СУБД временных рядов (InfluxDB)

## СУБД Ключ-значение:

- Redis
- Amazon DynamoDB
- Microsoft Azure Cosmos DB
- Memcached
- etcd

## Поисковые СУБД:

- Elasticsearch
- Splunk
- Solr
- MarkLogic
- Algolia

## Колоночные СУБД:

- Cassandra
- HBase
- Microsoft Azure Cosmos DB
- Datastax Enterprise
- Microsoft Azure Table Storage

# Что Такое Redis

**Redis** – система управления базами данных (СУБД) класса NoSQL с открытым исходным кодом, работающая со структурами данных типа «ключ – значение».

Используется для:

- Реализации кэшей,
- Баз данных key-value,
- Брокеров сообщений.



# Особенности Redis

Redis, как и всякая резидентная СУБД, хранит данные в оперативной памяти.

Другим примером со схожим функционалом является **Memcached**, но Redis имеет ряд особенностей и преимуществ.

Особенности Redis:

- Производительность;
- Персистентность;
- Структуры данных;
- Распределённость.





# Типы Данных в Redis

В отличие от Memcached, Redis поддерживает комплексные типы данных, что является одной из основных особенностей данной базы данных. Redis поддерживает 6 типов данных:

- 1) Строка;
- 2) Целое число;
- 3) Хэш-таблица (ассоциативный массив);
- 4) Двусвязный список;
- 5) Множество;
- 6) Упорядоченное множество;



# Подготовка к Установке Redis

Процесс установки зависит от используемой операционной системы:

- Windows;
- Linux;
- Mac OS;

**Важно:** Redis не поддерживает работу на операционной системе Windows. Для использования нужно установить WSL (Windows Subsystem Linux).

Если у Вас уже установлена Ubuntu, то Вы можете пропустить первые шаги.

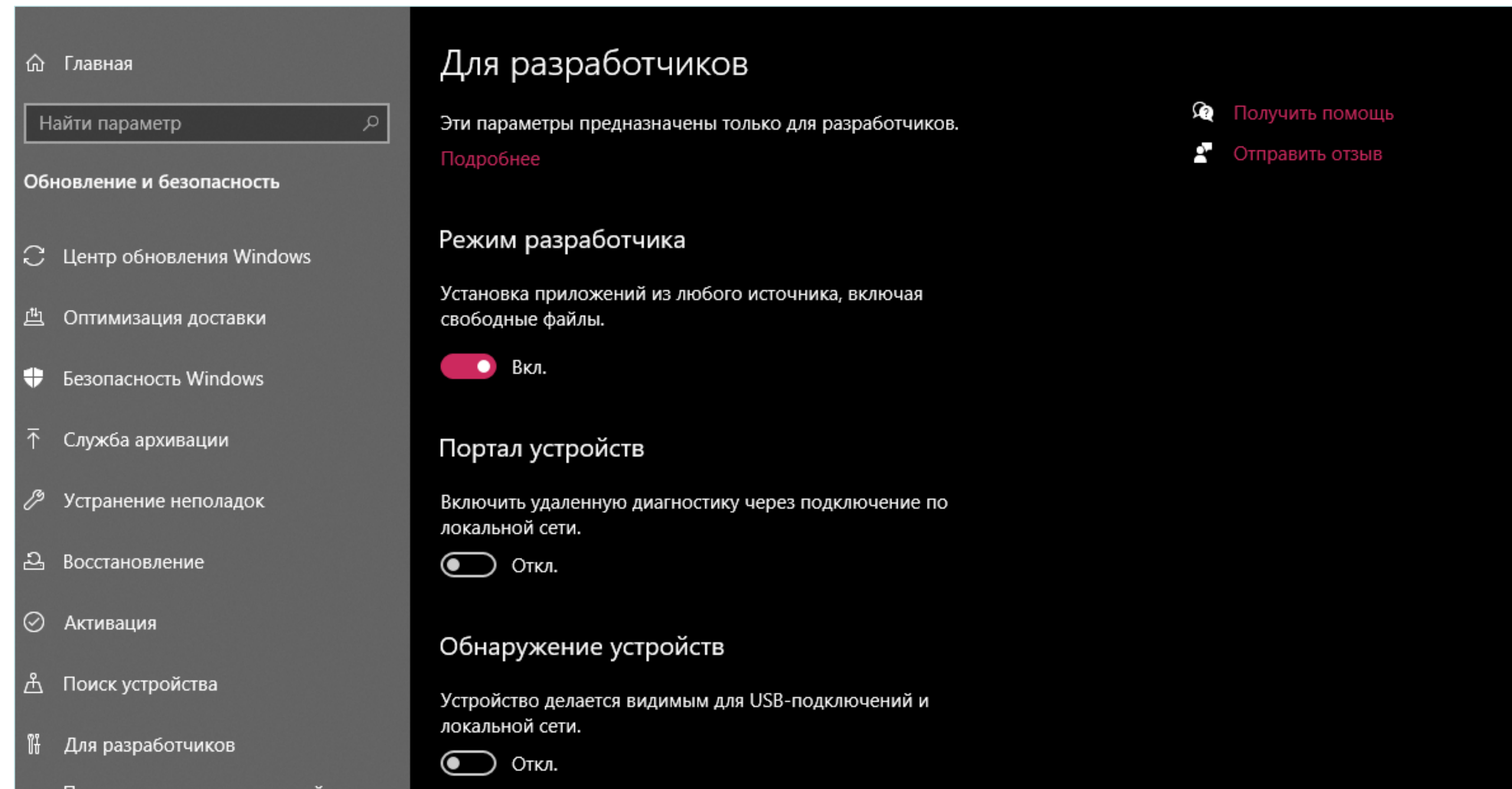
# Установка WSL и Переход в Режим Разработчика

Для установки Вам потребуется:

- 1) Включить режим разработчика;
- 2) Включить компонент: «Подсистема Windows для Linux»,
- 3) Установить «Ubuntu» из Microsoft Store.

Переход в режим разработчика:

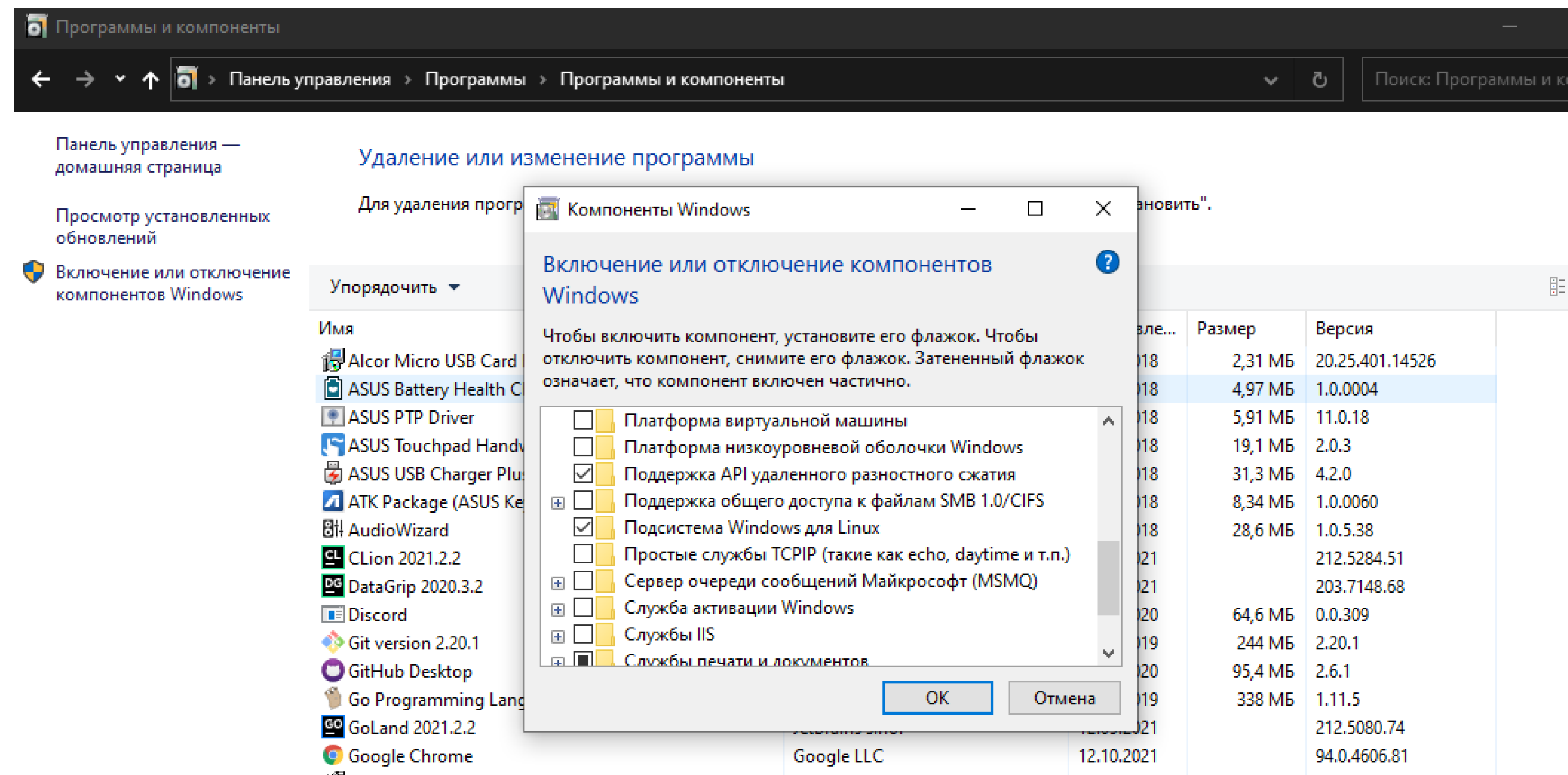
- Параметры
- Центр обновления Windows
- Для Разработчиков
- Режим разработчика.



# Включение «Подсистемы Windows для Linux»

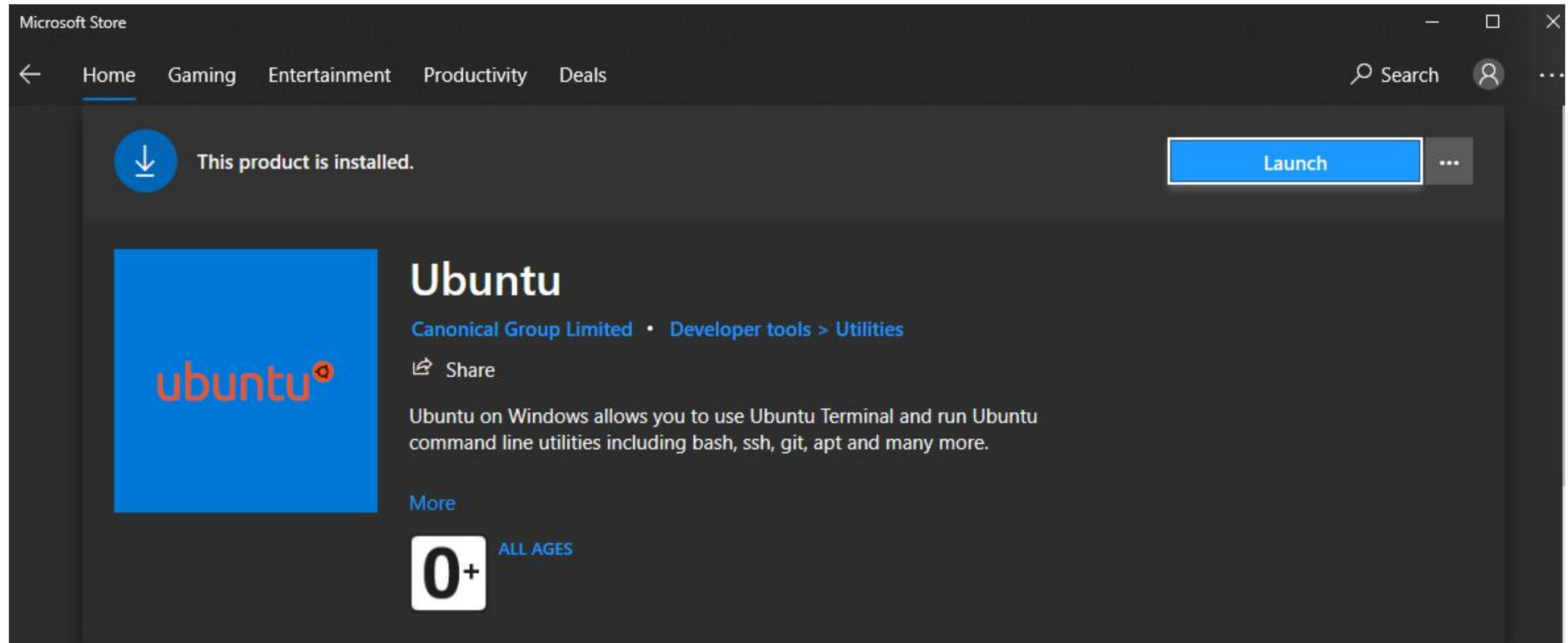
Для установки Вам потребуется:

- Панель управления
- Программы
- Программы и компоненты
- Включение или отключение компонентов Windows
- Подсистема Windows для Linux:



# Установка Ubuntu on Windows

Далее, необходимо установить Ubuntu через Microsoft Store:



# Установка Redis

Для установки Redis откройте терминал (*Ubuntu или WSL для Windows*) и введите следующие команды:

## 1) Для установки сервера Redis:

```
> sudo apt-get update  
> sudo apt-get upgrade  
> sudo apt-get install redis-server  
> redis-cli -v
```

## 2) Для перезапуска сервера Redis и проверки работоспособности:

```
> sudo service redis-server restart
```

## 3) Выполнение простой команды (использование клиента):

```
$ redis-cli  
127.0.0.1:6379> set user:1 «Jane»  
127.0.0.1:6379> get user:1  
«Jane»
```



# Redis Command Line Interface

**Redis Command Line Interface (CLI)** – способ взаимодействия с Redis сервером путём ввода команд в терминал.

Ниже представлен набор из нескольких простых команд:

```
goose@DESKTOP-0U8UV4T:/mnt/c/Users/Misha$ redis-cli
127.0.0.1:6379> set "Test string" "Hello"
OK
127.0.0.1:6379> get "Test string"
"Hello"
127.0.0.1:6379> set Name Misha EX 20
OK
127.0.0.1:6379> keys *
1) "Test string"
2) "Name"
127.0.0.1:6379> keys *
1) "Test string"
127.0.0.1:6379> FLUSHALL
OK
127.0.0.1:6379> keys *
(empty list or set)
```

```
127.0.0.1:6379> set Test "Hello"
OK
127.0.0.1:6379> get Test
"Hello"
127.0.0.1:6379> APPEND Test " world!"
(integer) 12
127.0.0.1:6379> get Test
"Hello world!"
127.0.0.1:6379> set counter 3
OK
127.0.0.1:6379> INCR counter
(integer) 4
127.0.0.1:6379> DECR counter
(integer) 3
127.0.0.1:6379> getset Test "My name is Redis"
"Hello world!"
127.0.0.1:6379> get Test
"My name is Redis"
127.0.0.1:6379> _
```

# Хэш-Таблица

**Хэш-таблица** (ассоциативный массив) – структура данных, которая хранит пары «ключ – значение». В BCL хэш-таблица реализована классом

**Dictionary<T key, V value>**.

```
goose@DESKTOP-0U8UV4T: /mnt/c/Users/Misha
goose@DESKTOP-0U8UV4T:/mnt/c/Users/Misha$ redis-cli
127.0.0.1:6379> HSET person name Misha
(integer) 1
127.0.0.1:6379> HSET person surname Andryuschenko
(integer) 1
127.0.0.1:6379> HSET person age 20
(integer) 1
127.0.0.1:6379> HGET person age
"20"
127.0.0.1:6379> HGETALL person
1) "name"
2) "Misha"
3) "surname"
4) "Andryuschenko"
5) "age"
6) "20"
127.0.0.1:6379> HVALS person
1) "Misha"
2) "Andryuschenko"
3) "20"
127.0.0.1:6379> HKEYS person
1) "name"
2) "surname"
3) "age"
127.0.0.1:6379> _
```



# Множество

**Множество** – структура данных, которая хранит значения в единственном экземпляре.

Redis поддерживает стандартные операции с множествами: *объединение, пересечение, симметрическую разность, кардинальное число (мощность)* и другие.

В BCL множество реализовано классом [HashSet<T>](#).

```
goose@DESKTOP-0U8UV4T:/mnt/c/Users/Misha$ redis-cli
127.0.0.1:6379> SADD persons Misha Anna Kate Dima
(integer) 4
127.0.0.1:6379> SMEMBERS persons
1) "Misha"
2) "Dima"
3) "Kate"
4) "Anna"
127.0.0.1:6379> SADD another Kate Eva Liza
(integer) 3
127.0.0.1:6379> SMEMBERS another
1) "Liza"
2) "Eva"
3) "Kate"
127.0.0.1:6379> SCARD another
(integer) 3
```

```
127.0.0.1:6379> SUNION persons another
1) "Anna"
2) "Kate"
3) "Misha"
4) "Dima"
5) "Eva"
6) "Liza"
127.0.0.1:6379> SINTER persons another
1) "Kate"
127.0.0.1:6379> SDIFF persons another
1) "Misha"
2) "Dima"
3) "Anna"
127.0.0.1:6379>
```

# Двусвязный Список

**Список** (в данном случае – двусвязный) – структура данных, для которой добавление и удаление в конец/начало имеет асимптотическую сложность  $O(1)$ .

В BCL двусвязный список реализован классом [LinkedList<T>](#).

```
goose@DESKTOP-0U8UV4T:/mnt/c/Users/Misha$ redis-cli
127.0.0.1:6379> LPUSH persons Misha Vanya Kate
(integer) 3
127.0.0.1:6379> LRANGE person 0 1
(empty list or set)
127.0.0.1:6379> LRANGE persons 0 1
1) "Kate"
2) "Vanya"
127.0.0.1:6379> LRANGE persons 0 -1
1) "Kate"
2) "Vanya"
3) "Misha"
127.0.0.1:6379> RPUSH persons Liza
(integer) 4
127.0.0.1:6379> LRANGE persons 0 -1
1) "Kate"
2) "Vanya"
3) "Misha"
4) "Liza"
127.0.0.1:6379> LPOP persons
"Kate"
127.0.0.1:6379> RPOP persons
"Liza"
127.0.0.1:6379> LLEN persons
(integer) 2
127.0.0.1:6379>
```

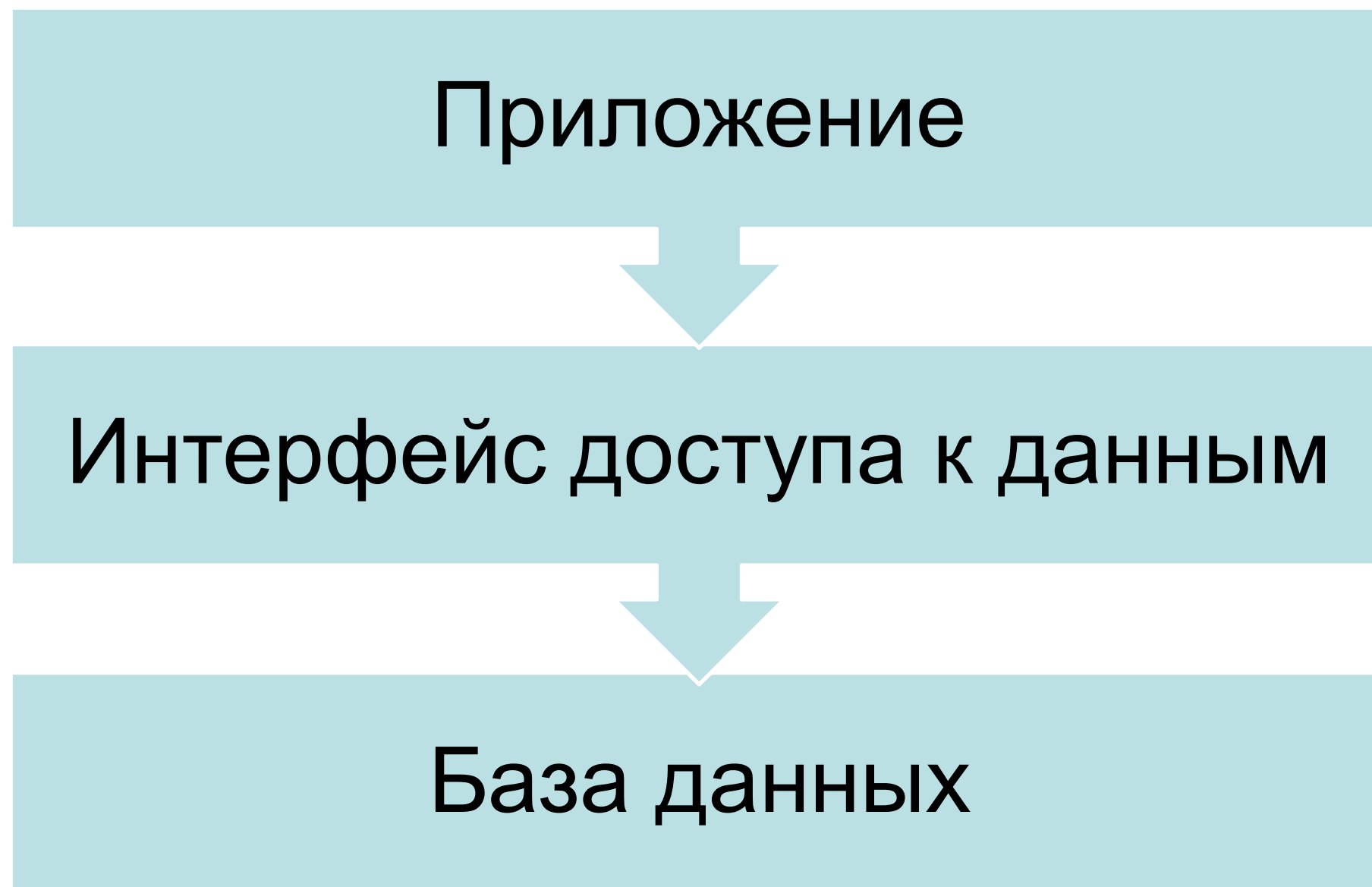
# Упорядоченное Множество

**Упорядоченное множество** – множество, упорядоченное по значению ключа. Доступ к элементам множества имеет асимптотическую сложность  $O(\log n)$ .

В BCL упорядоченное множество реализовано классом [SortedSet<T>](#).

```
goose@DESKTOP-0U8UV4T: /mnt/c/Users/Misha
goose@DESKTOP-0U8UV4T:/mnt/c/Users/Misha$ redis-cli
127.0.0.1:6379> ZADD persons 2001 Misha
(integer) 1
127.0.0.1:6379> ZADD persons 2000 Dima
(integer) 1
127.0.0.1:6379> ZADD persons 2003 Eva
(integer) 1
127.0.0.1:6379> ZRANGE persons 0 -1
1) "Dima"
2) "Misha"
3) "Eva"
127.0.0.1:6379> ZRANGE persons 0 -1 withscores
1) "Dima"
2) "2000"
3) "Misha"
4) "2001"
5) "Eva"
6) "2003"
127.0.0.1:6379> ZRANK persons Misha
(integer) 1
127.0.0.1:6379> ZRANGE persons 1 1 withscores
1) "Misha"
2) "2001"
127.0.0.1:6379> _
```

# Работа с базами данных



## Библиотеки для работы с Redis из .Net (C#):

- StackExchange.Redis
- ServiceStack.Redis
- BeetleX.Redis
- FreeRedis
- NewLife.Redis
- Nhiredis
- Rediska

## Клиенты для работы с Redis:

<https://redis.io/clients>

ActionScript	ActiveX/COM+	Bash	Boomi	C	C#
C++	Clojure	Common Lisp	Crystal	D	Dart
Delphi	Elixir	emacs lisp	Erlang	Fancy	gawk
GNU Prolog	Go	Haskell	Haxe	Io	Java
Julia	Lasso	Lua	Matlab	mruby	Nim
Node.js	Objective-C	OCaml	Pascal	Perl	PHP
PL/SQL	Prolog	Pure Data	Python	R	Racket
Rebol	Ruby	Rust	Scala	Scheme	Smalltalk
Swift	Tcl	VB	VCL	Xojo	Zig



Для использования сторонних библиотек необходим NuGet

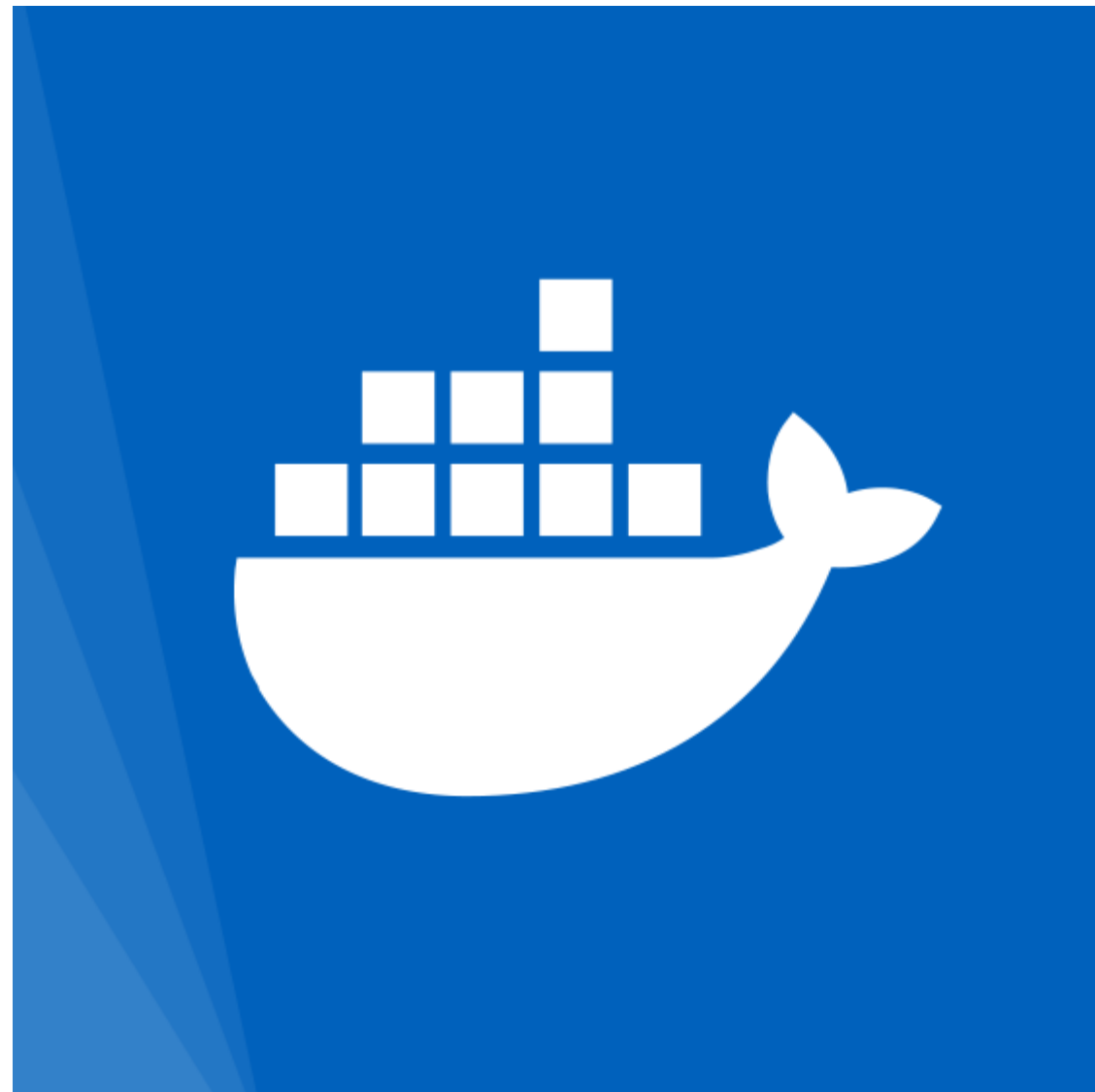
**NuGet** – это система управления пакетами для .NET (встроен в VS).

<https://www.nuget.org/>

# Альтернатива: Redis из под Docker

## Установка Docker

---



**Docker** — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений. Позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть развёрнут на Linux-совместимой системе. Предоставляет набор команд для управления этими контейнерами.

<https://hub.docker.com/editions/community/docker-ce-desktop-windows>

### Требования:

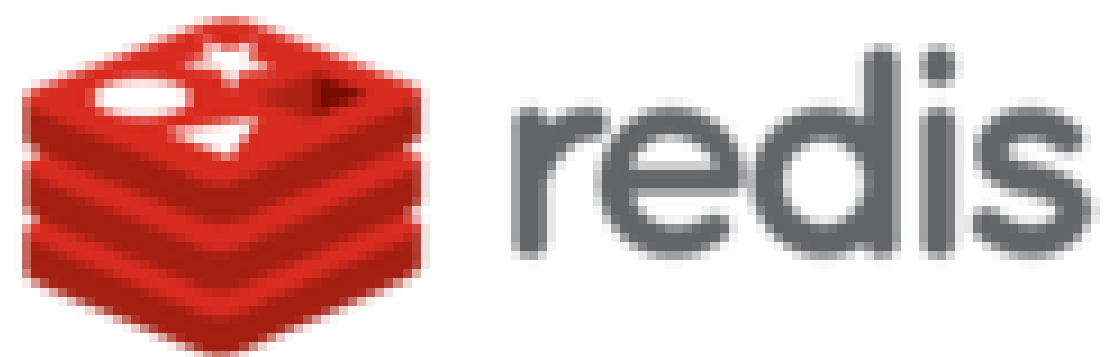
Microsoft Windows 10 Professional или Enterprise 64-bit, или Windows 10 Home 64-bit с поддержкой WSL 2.

# Альтернатива: Redis из под Docker


## Установка Redis с Docker

---

[https://hub.docker.com/\\_/redis](https://hub.docker.com/_/redis)



**В командной строке:**  
`docker pull redis`

 Командная строка

```
Microsoft Windows [Version 10.0.19043.1320]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\vic>docker pull redis
Using default tag: latest
latest: Pulling from library/redis
7d63c13d9b9b: Pull complete
a2c3b174c5ad: Pull complete
283a10257b0f: Pull complete
7a08c63a873a: Pull complete
0531663a7f55: Pull complete
9bf50efb265c: Pull complete
Digest: sha256:a89cb097693dd354de598d279c304a1c73ee550fbfff6d9ee515568e0c749cfe
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest

C:\Users\vic>
```

# Альтернатива: Redis из под Docker

## Запуск контейнера с Redis и проброс портов



<https://www.ionos.com/digitalguide/server/know-how/redis-in-docker-containers/>

**В командной строке:**

```
docker run --name my-redis-container -p 7001:6379 -d redis
```

```
C:\Users\vic>docker run --name my-redis-container -p 7001:6379 -d redis  
b6e1d87745a2f70b5b97e58b2c8cfb1c1498d05b37c2529fc2fc669bb3ea50ae
```

A screenshot of the Docker Desktop application window. The window has a blue header bar with the Docker logo, an "Upgrade" button, and user information "vicdudarev". On the left is a sidebar with navigation options: "Containers / Apps", "Images", "Volumes", and "Dev Environments" (with a "PREVIEW" button). The main area displays details for a container with ID "4245c0a7d40e643205c21e0681bdbdfbfb7ed94183...". It shows the container is "IN USE" and was "CREATED 3 minutes ago". Below this is a table with the following data:

CONTAINER NAME	IMAGE	PORT	TARGET
my-redis-container	redis	7001	/data

The container name "my-redis-container" is underlined with a red line, and the port "7001" is circled with a red line.



## Использование контейнера с Redis

---



<https://www.ionos.com/digitalguide/server/know-how/redis-in-docker-containers/>

**В командной строке WSL:**

```
redis-cli -h 127.0.0.1 -p 7001
```

```
vic@VIC-NOTE2:/mnt/c/WINDOWS/system32$ redis-cli -h 127.0.0.1 -p 7001
```

```
127.0.0.1:7001> get test
```

```
(nil)
```

```
127.0.0.1:7001> set test 15
```

```
OK
```

```
127.0.0.1:7001> get test
```

```
"15"
```