Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Архитектура вычислительных систем

*К защите допустить*:  
И. О. заведующего кафедрой информатики  
\_\_\_\_\_\_\_\_\_\_\_\_ С. И. Сиротко

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

**ШАХМАТЫ ПО BLUETOOTH**

БГУИР КП 1-40 04 01 031 ПЗ

Студент М. А. Щур

Руководитель С. И. Сиротко

Нормоконтролер С. И. Сиротко

Минск 2024

# **СОДЕРЖАНИЕ**

[Содержание 4](#_Toc191478546)

[1 Архитектура программного обеспечения 5](#_Toc191478547)

[1.1 Описание используемых технологий 5](#_Toc191478548)

[1.2 История, версии и достоинства 7](#_Toc191478549)

[2 Платформа программного обеспечения 10](#_Toc191478550)

[2.1 Операционная система 10](#_Toc191478551)

[2.2 Язык и среда разработки 13](#_Toc191478552)

[2.3 История, версии и достоинства 16](#_Toc191478553)

[Заключение 18](#_Toc191478554)

[Список литературных источников 19](#_Toc191478555)

# 

# **1 АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

## **1.1 Описание используемых технологий**

*Bluetooth* – это стандарт беспроводной связи, который позволяет устройствам обмениваться данными на коротких расстояниях. Он работает в диапазоне частот 2.4 ГГц и обеспечивает связь на расстоянии до 10 метров, хотя для некоторых устройств этот диапазон может достигать 100 метров. *Bluetooth* стал основным способом подключения различных устройств, включая смартфоны, компьютеры, наушники, колонки и устройства для умного дома.

*Bluetooth* использует несколько протоколов для передачи данных, включая *L*2*CAP* (*Logical* *Link* *Control* *and* *Adaptation* *Protocol*), *RFCOMM* (*Serial* *Port* *Emulation*), *SDP* (*Service* *Discovery* *Protocol*) и другие. Эти протоколы обеспечивают надежную передачу данных, а также совместимость между устройствами разных производителей. *Bluetooth* поддерживает различные профили, которые определяют, как устройства могут взаимодействовать друг с другом. Например, профиль *A*2*DP* (*Advanced* *Audio* *Distribution* *Profile*) позволяет передавать стереозвук между устройствами, а профиль *HFP* (*Hands*-*Free* *Profile*) используется для подключения гарнитур. На рисунке 1.1 изображена схема архитектуры *Bluetooth*.

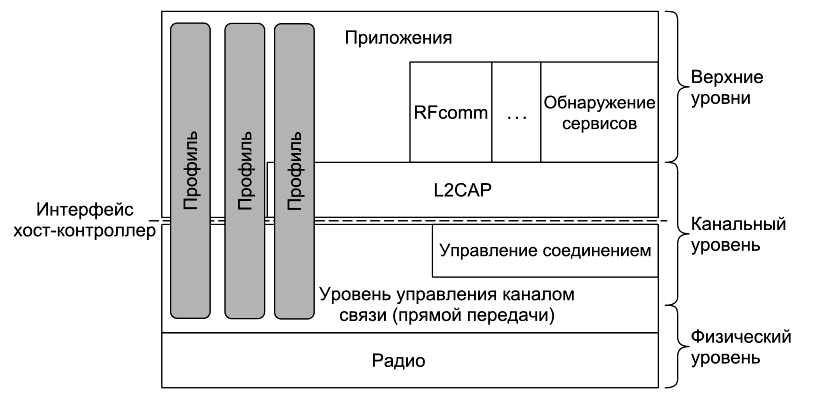


Рисунок 1.1 – Схема архитектуры *Bluetooth*

*Bluetooth* *Low* *Energy* (*BLE*), представленная в версии 4.0, значительно улучшила энергоэффективность технологии. *BLE* разработан с акцентом на минимальное потребление энергии, что позволяет устройствам работать гораздо дольше без подзарядки. Это особенно важно для носимых технологий, таких как фитнес-трекеры и медицинские устройства, которые собирают данные о здоровье и активности пользователей. *BLE* позволяет передавать небольшие объемы данных с минимальными затратами энергии, что делает его идеальным для приложений, требующих постоянного подключения.

*Bluetooth* находит широкое применение в различных областях:

1 Мобильные устройства. *Bluetooth* используется для подключения гарнитур, беспроводных клавиатур, мышей и колонок. Это позволяет пользователям наслаждаться свободой движения и удобством использования без проводов.

2 Умный дом. Технология позволяет управлять умными устройствами, такими как освещение, термостаты, замки и камеры видеонаблюдения. Подключение через *Bluetooth* создает интегрированные системы, которые можно контролировать с помощью смартфонов или голосовых помощников, таких как *Amazon* *Alexa* и *Google* *Assistant*.

3 Носимые устройства. *Bluetooth* широко используется в фитнес-трекерах, умных часах и других носимых гаджетах. Эти устройства собирают и передают данные о здоровье, такие как частота сердечных сокращений, уровень активности и качество сна, что позволяет пользователям отслеживать свое состояние и улучшать физическую форм

4 Автомобили. *Bluetooth* обеспечивает беспроводное соединение мобильных устройств с автомобильными системами. Это позволяет пользователям управлять мультимедийной системой, принимать звонки и использовать навигацию, не отвлекаясь от дороги. Многие автомобили теперь оснащены встроенными системами *Bluetooth*, которые обеспечивают полную интеграцию со смартфонами.

5 Интернет вещей (*IoT*). *Bluetooth* играет важную роль в экосистеме *IoT*, позволяя различным устройствам обмениваться данными и взаимодействовать в рамках единой сети. Это включает в себя датчики, термометры, системы безопасности и другие устройства, которые могут обмениваться данными для создания умных домов и городов.

*Bluetooth* включает несколько уровней безопасности, чтобы защитить передаваемые данные. Это включает аутентификацию, которая подтверждает идентичность устройств, и шифрование, которое защищает данные от несанкционированного доступа. Пользователи могут управлять режимами видимости своих устройств, что позволяет им выбирать, кто может их обнаружить. Однако, несмотря на эти меры, *Bluetooth* все еще подвержен некоторым уязвимостям, таким как атаки «человек посередине» и другие виды злоупотреблений, что подчеркивает важность соблюдения лучших практик безопасности.

С каждым обновлением *Bluetooth* становится более совершенным. Ожидается, что будущие версии будут улучшены в плане скорости передачи данных, радиуса действия и возможности позиционирования. Например, *Bluetooth* 5.1 добавляет функции, такие как угловая локализация, что позволяет более точно определять местоположение устройств. Это открывает новые возможности для приложений в области навигации, логистики и умного дома.

Еще одним важным направлением является интеграция *Bluetooth* с другими технологиями беспроводной связи. Например, *Bluetooth* может использоваться в сочетании с *Wi*-*Fi* для улучшения передачи данных и расширения диапазона действия. Это позволит создавать более сложные и мощные системы, которые могут адаптироваться к требованиям пользователей.

*Bluetooth* представляет собой мощную и универсальную технологию, которая продолжает эволюционировать и адаптироваться к современным требованиям. С помощью *Bluetooth* возможно создание удобных и эффективных решений для различных задач, начиная от простого подключения аксессуаров и заканчивая сложными системами автоматизации и управления умным домом. В современном мире, где беспроводные технологии становятся все более популярными, *Bluetooth* остается ключевым игроком, обеспечивая надежную и безопасную связь между устройствами.

## **1.2 История, версии и достоинства**

*Bluetooth* был разработан в 1994 году компанией *Ericsson* с целью создания беспроводного стандарта для замены проводных соединений между мобильными устройствами и аксессуарами. В то время, когда технологии связи развивались стремительными темпами, возникла необходимость в более удобном и эффективном способе подключения устройств без использования проводов.

Первоначально проект носил имя «*Bluetooth*» в честь датского короля Харальда «Синезубого» Грея, который известен тем, что объединил Данию и Норвегию в *X* веке. Это название символизировало идею объединения различных устройств в одну сеть, что стало ключевым аспектом философии *Bluetooth*. В 1998 году был основан консорциум *Bluetooth* *Special* *Interest* *Group* (*SIG*), который объединил ведущие компании, такие как *IBM*, *Intel*, *Nokia* и *Toshiba*, для дальнейшего развития и стандартизации технологии. Существует множество версий *Bluetooth* среди которых:

1 *Bluetooth* 1.0 и 1.1. Первая версия *Bluetooth*, выпущенная в 1999 году, имела множество проблем с совместимостью и производительностью. Однако, несмотря на это, она положила начало развитию стандарта. *Bluetooth* 1.0 предоставил основные функции беспроводной связи, такие как возможность обмена данными на расстоянии до 10 метров. Версия 1.1, выпущенная в 2001 году, исправила многие изначальные недостатки и добавила поддержку различных профилей, что увеличило совместимость с устройствами разных производителей.

2 *Bluetooth* 2.0 и *Enhanced* *Data* *Rate* (*EDR*). В 2004 году была представлена версия 2.0, которая включала *Enhanced* *Data* *Rate* (*EDR*), обеспечивающий более высокую скорость передачи данных – до 3 Мбит/с. Это стало значительным шагом вперед, так как позволило улучшить качество передачи аудио и видео. *Bluetooth* 2.0 также улучшил энергопотребление и обеспечил большую стабильность соединения.

3 *Bluetooth* 3.0 и *High* *Speed*. В 2009 году была выпущена версия 3.0, которая представила возможность передачи данных на высоких скоростях (*High* *Speed*) с использованием *Wi*-*Fi*. Это позволяло *Bluetooth* передавать большие объемы данных быстрее, чем когда-либо прежде. Версия 3.0 также улучшила взаимодействие между устройствами, предоставляя более надежные соединения и поддержку новых профилей.

4 *Bluetooth* 4.0 и *Low* *Energy*. 2010 год стал важной вехой с выходом *Bluetooth* 4.0, который представил *Bluetooth* *Low* *Energy* (*BLE*). Эта версия была разработана для устройств с низким потреблением энергии, таких как фитнес-трекеры и медицинские датчики. *BLE* позволяет устройствам работать дольше от батареи, передавая небольшие объемы данных с минимальными затратами энергии. Это открывает новые возможности для носимых технологий и интернета вещей (*IoT*).

5 *Bluetooth* 4.1 и 4.2. В 2013 году была выпущена версия 4.1, которая улучшила совместимость с *LTE*-сетями и обеспечила более эффективное использование радиочастотного спектра. *Bluetooth* 4.2, представленный в 2014 году, добавил поддержку *IPv*6, что позволило устройствам напрямую подключаться к интернету, улучшив взаимодействие в рамках *IoT*. Также была улучшена безопасность передачи данных и увеличена скорость передачи данных в *BLE*.

6 *Bluetooth* 5.0 и улучшения. В 2016 году была представлена версия 5.0, которая значительно увеличила радиус действия (до 240 метров) и скорость передачи данных (до 2 Мбит/с) по сравнению с предыдущими версиями. Эта версия также поддерживает возможность подключения к большему числу устройств одновременно, что особенно важно для экосистемы *IoT*. *Bluetooth* 5.0 улучшил функции позиционирования, что открыло новые горизонты для приложений в логистике и навигации.

7 *Bluetooth* 5.1 и 5.2. *Bluetooth* 5.1, выпущенный в 2019 году, добавил новые функции, такие как угловая локализация, позволяющая более точно определять местоположение устройств в пространстве. Это значительно улучшило возможности навигации и взаимодействия между устройствами. В 2020 году была представлена версия 5.2, которая включала такие функции, как *LE* *Audio*, улучшение качества звука и поддержку новых профилей, таких как *Audio* *Streaming* и *Broadcast* *Audio*, что позволяет передавать аудио на несколько устройств одновременно.

8 *Bluetooth* 5.3 и будущее. На момент 2021 года *Bluetooth* 5.3 была представлена как последняя версия, которая улучшила управление энергопотреблением и расширила функциональность существующих профилей. Ожидается, что будущие версии *Bluetooth* будут продолжать развиваться в направлении повышения скорости передачи данных, увеличения радиуса действия и улучшения взаимодействия между устройствами.

В использовании технологии *Bluetooth* есть следующие преимущества:

1 Удобство. *Bluetooth* позволяет легко подключать и отключать устройства без проводов, что делает его идеальным выбором для мобильных пользователей.

2 Гибкость. Поддерживает множество устройств и профилей, что позволяет использовать его в самых разных сценариях.

3 Энергоэффективность. *BLE* обеспечивает длительное время работы от батареи для носимых устройств, что является важным фактором для пользователей.

Однако существуют и недостатки в использовании *Bluetooth*.

1 Ограниченная дальность действия. На расстоянии более 10 метров связь может ухудшаться, что ограничивает возможности использования в больших помещениях.

2 Скорость передачи данных. Скорость передачи данных *Bluetooth* ниже, чем у некоторых других технологий, таких как *Wi*-*Fi*, что может быть недостатком для приложений, требующих высокой скорости.

3 Помехи. Работа на частоте 2.4 ГГц может вызывать помехи от других беспроводных устройств, таких как микроволновые печи и *Wi*-*Fi* маршрутизаторы.

*Bluetooth* действительно прошел значительный путь с момента своего появления в конце 1990-х годов. Изначально разработанный для замены проводных соединений между устройствами, *Bluetooth* сегодня стал важной частью нашей повседневной жизни. Первые версии, такие как *Bluetooth* 1.0 и 1.1, обеспечивали базовую связь на коротких расстояниях, но имели множество проблем с совместимостью и стабильностью соединения. С выходом *Bluetooth* 2.0 и 2.1 были улучшены скорость передачи данных и безопасность, благодаря введению технологии *Enhanced* *Data* *Rate* (*EDR*), что значительно увеличивало скорость передачи. *Bluetooth* 3.0 добавил возможность передачи данных по *Wi*-*Fi*, а *Bluetooth* 4.0 ввел технологию *Low* *Energy* (*BLE*), что позволило устройствам работать дольше на батарейках, особенно для *IoT*-устройств. Последние версии, такие как *Bluetooth* 5.0 и выше, предложили еще более высокую скорость, увеличенную дальность действия и возможность подключения к нескольким устройствам одновременно, открывая новые возможности для разработки умных технологий, таких как умные дома и носимые устройства. *Bluetooth* находит применение в самых разных сферах: от синхронизации смартфонов с наушниками и колонками до управления бытовой техникой через смартфон, а также мониторинга состояния здоровья с помощью носимых устройств. С развитием технологий и увеличением числа *IoT*-устройств, *Bluetooth* продолжит адаптироваться к новым требованиям. Ожидается, что будущие версии будут фокусироваться на улучшении безопасности, повышении скорости передачи данных и расширении диапазона действия, что сделает технологии еще более доступными и удобными для пользователей. *Bluetooth* остаётся неотъемлемой частью экосистемы беспроводной связи, и его развитие будет способствовать созданию ещё более инновационных решений в области технологий.

# **2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

## **2.1 Операционная система**

В качестве операционной системы была выбрана *Linux*.*Linux* – это свободная и открытая операционная система, которая служит платформой для множества приложений и инструментов. Она была создана Линусом Торвальдсом в 1991 году и с тех пор стала одной из самых популярных операционных систем в мире.

Первоначально Торвальдс не хотел продавать свою разработку и запрещал это другим. В уведомлении об авторских правах в файле *COPYING* первой версии (0.01) он установил жесткие ограничения на распространение *Linux*. Но в феврале 1992 года пользователи начали обращаться к нему с просьбой разрешить взимать плату за распространение дискет с *Linux*, чтобы покрыть затраты. Также важно учитывать, что при создании *Linux* использовались различные свободно распространяемые инструменты, самым значимым из которых является компилятор *GCC*, лицензированный под *GPL*, разработанной Ричардом Столлманом. В результате Торвальдсу пришлось изменить свои условия, и с версии 0.12 он также стал использовать *GPL*. С технической точки зрения, *Linux* представляет собой ядро *Unix*-подобной операционной системы, отвечающее за взаимодействие с аппаратной частью компьютера, распределение памяти и выделение процессорного времени программам.

Операционная система включает множество утилит, обеспечивающих взаимодействие пользователя с системой. Успех *Linux* во многом обусловлен наличием утилит, разработанных в рамках проекта *GNU* к 1991 году. Проекту *GNU* не хватало ядра, и, скорее всего, *Linux* остался бы невостребованным без необходимых утилит. Ричард Столлман прав, настаивая на том, что операционную систему следует называть не только *Linux*, но и *GNU*/*Linux*. Тем не менее, название *Linux* стало исторически закрепленным за этой ОС. *Linux* является модульной и гибкой платформой, поддерживающей различные архитектуры и аппаратное обеспечение. Он может работать на серверах, настольных компьютерах, мобильных устройствах, встроенных системах и суперкомпьютерах. Благодаря этой универсальности, *Linux* нашел применение в таких областях, как веб-серверы, базы данных, научные исследования, разработка программного обеспечения, мультимедиа и мобильные устройства. *Linux* обладает мощной командной строкой, которая предоставляет множество инструментов для управления системой и выполнения задач.

Существуют также графические среды разработки (*IDE*) и интерфейсы пользователя, делающие работу с *Linux* более удобной для начинающих. Кроме того, *Linux* поддерживает огромное количество программного обеспечения. Существует множество приложений для различных задач, включая офисные пакеты, браузеры, мультимедийные инструменты, графические редакторы и многое другое. Большинство из них доступны бесплатно и имеют открытый исходный код. В целом, *Linux* представляет собой мощную и гибкую платформу программного обеспечения, обеспечивающую большую свободу и контроль над компьютерной системой. Благодаря своей открытой природе и активному сообществу разработчиков, *Linux* продолжает развиваться и оставаться одним из главных игроков в мире операционных систем.

Логический диск в *Linux* представляет собой раздел или том на физическом диске. Разделы позволяют разделить физический диск на несколько изолированных областей, каждая из которых может иметь свою файловую систему. Логические диски и их разделы представлены в виде файлов в каталоге /*dev*. Для работы с логическими дисками в *Linux* используются утилиты, такие как *fdisk*, *parted* и *lsblk*. *fdisk* и *parted* позволяют создавать, изменять и удалять разделы на физическом диске, а *lsblk* выводит информацию о физических и логических дисках, а также о разделах.

Логические диски могут быть отформатированы с использованием различных файловых систем, таких как *ext*4, *XFS*, *Btrfs* и другие. Форматирование диска создает файловую систему, позволяющую хранить файлы и каталоги.

Разделы являются более мелкими частями всего диска. В *Linux* они обозначаются с помощью цифры после названия блочного устройства и, следовательно, получают такие имена, как, например, /*dev*/*sda*1 и /*dev*/*sdb*3. Ядро представляет каждый раздел в виде блочного устройства, как если бы это был целый диск. Хотя ядро и позволяет иметь одновременный доступ ко всему диску и к одному из его разделов, этого не придется делать, если только нет цели скопировать весь диск. Если пользователю необходим доступ к данным в файле, ему потребуется выяснить из таблицы разделов расположение соответствующего раздела, а затем отыскать в базе данных файловой системы этого раздела желаемый файл с данными. Чтобы обращаться к данным на диске, ядро *Linux* использует систему слоев, показанную на рисунке 2.1. Следует обратить внимание на то, что с дисками можно работать как с помощью файловой системы, так и непосредственно через дисковые устройства.

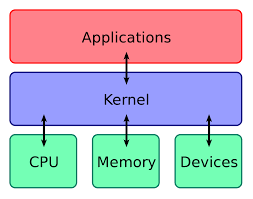


Рисунок 2.1 – Система слоёв *Linux*

Существуют различные типы таблиц разделов. Традиционная таблица – та, которая расположена внутри главной загрузочной записи *MBR* (*Master* *Boot* *Record*). Новым, набирающим силу стандартом является глобальная таблица разделов с уникальными идентификаторами *GPT* (*Globally* *Unique* *Identifier* *Partition* *Table*). Просмотр таблиц разделов – операция сравнительно простая и безвредная. Изменение таблиц разделов также осуществляется довольно просто, однако при таком типе изменений диска могут возникнуть опасности:

1 Изменение таблицы разделов сильно усложняет восстановление любых данных в удаляемых разделах, поскольку при этом меняется начальная точка привязки файловой системы. Обязательно нужно создавать резервную копию диска, на котором меняются разделы, если он содержит важную информацию.

2 Следует убедится в том, что на целевом диске ни один из разделов в данный момент не используется. Это важно, поскольку в большинстве версий *Linux* автоматически монтируется любая обнаруженная файловая система.

Последним звеном между ядром и пространством пользователя для дисков обычно является файловая система. С ней помогают взаимодействовать, такие команды, как *ls* и *cd*. Как отмечалось ранее, файловая система является разновидностью базы данных; она поддерживает структуру, призванную трансформировать простое блочное устройство в замысловатую иерархию файлов и подкаталогов, которую пользователи способны понять.

В свое время файловые системы, располагавшиеся на дисках и других физических устройствах, использовались исключительно для хранения данных. Однако древовидная структура каталогов, а также интерфейс ввода вывода довольно гибки, поэтому теперь файловые системы выполняют множество задач, например роль системных интерфейсов. Файловые системы традиционно реализованы внутри ядра, однако инновационный протокол 9*P* из операционной системы *Plan* 9 способствовал разработке файловых систем в пространстве пользователя.

В *Linux* включена поддержка таких файловых систем, как «родные» разработки, оптимизированные для *Linux*, «чужеродные» типы, например семейство *Windows* *FAT*, универсальные файловые системы вроде *ISO* 9660 и множество других. В приведенном ниже списке перечислены наиболее распространенные типы файловых систем для хранения данных. Имена типов систем, как их определяет *Linux*, приведены в скобках после названия файловых систем.

1 Четвертая расширенная файловая система (*ext*4) является текущей реализацией в линейке «родных» для *Linux* файловых систем. Вторая расширенная файловая система (*ext*2) долгое время была системой по умолчанию в системах *Linux*, которые испытывали влияние традиционных файловых систем *Unix*, таких как файловая система *Unix* (*UFS*, *Unix* *File* *System*) и быстрая файловая система (*FFS*, *Fast* *File* *System*). В третьей расширенной файловой системе (*ext*3) появился режим журналирования (небольшой кэш за пределами нормальной структуры данных файловой системы) для улучшения целостности данных и ускорения загрузки системы. Файловая система *ext*4 является дальнейшим улучшением, с поддержкой файлов большего размера по сравнению с допустимым в системах *ext*2 или *ext*3, а также большего количества подкаталогов. Среди расширенных файловых систем присутствует некоторая доля обратной совместимости. Например, можно смонтировать систему *ext*2 как *ext*3 или наоборот, а также смонтировать файловые системы *ext*2 и *ext*3 как *ext*4, однако нельзя смонтировать файловую систему *ext*4 как *ext*2 или *ext*3.

2 Файловая система *ISO* 9660 (*iso*9660) – это стандарт для дисков *CD* *ROM*. Большинство дисков *CD*-*ROM* использует какой-либо вариант стандарта *ISO* 9660.

3 Файловые системы *FAT* (*msdos*, *vfat*, *umsdos*) относятся к системам *Microsoft*. Простой тип *msdos* поддерживает весьма примитивное унылое многообразие систем *MS*-*DOS*. Для большинства современных файловых систем *Windows* следует использовать тип *vfat*, чтобы получить возможность полного доступа из *OC* *Linux*. Редко используемый тип *umsdos* представляет интерес для *Linux*: в нем есть поддержка таких особенностей *Unix*, как символические ссылки, которые находятся над файловой системой *MS*-*DOS*. 4 Тип *HFS*+ (*hfsplus*) является стандартом *Apple*, который используется в большинстве компьютеров *Macintosh*. Хотя расширенные файловые системы были абсолютно пригодны для применения обычными пользователями, в технологии файловых систем были произведены многочисленные улучшения, причем такие, что даже система *ext*4 не может ими воспользоваться в силу требований обратной совместимости.

## **2.2 Язык и среда разработки**

Выбор языка программирования *C*# и среды разработки *Visual* *Studio* для реализации программного обеспечения по теме данного проекта обоснован рядом преимуществ, которые предоставляют эти технологии.

*C*# – это объектно-ориентированный язык программирования, разработанный компанией *Microsoft*, который широко используется для создания разнообразных приложений, включая веб-приложения, десктопные приложения, игры и многое другое. Некоторые из его преимуществ:

1 Мощная типизация и безопасность. *C*# предоставляет строгую типизацию, что обеспечивает более безопасное программирование и предотвращает множество типичных ошибок во время выполнения программы. Кроме того, язык обладает механизмом обработки исключений, позволяющим элегантно обрабатывать исключительные ситуации.

2 Богатые возможности для разработки приложений. *C*# и платформа .*NET* предоставляют широкий спектр инструментов и библиотек для разработки приложений различных типов: от десктопных приложений до веб-приложений и мобильных приложений. Это включает в себя *ASP*.*NET* для веб-разработки, *WPF* и *WinForms* для десктопных приложений, *Xamarin* для мобильной разработки и многое другое.

3 Многоплатформенность и переносимость. Платформа .*NET* является многоплатформенной, что означает, что приложения, написанные на *C*#, могут быть запущены на различных операционных системах, таких как *Windows*, *Linux* и *macOS*. Это обеспечивает большую гибкость и переносимость для разработанных приложений.

4 Интеграция с другими технологиями *Microsoft*. *C*# тесно интегрирован с другими технологиями и продуктами *Microsoft*, такими как базы данных *SQL* *Server*, облачная платформа *Azure*, средства разработки игр на движке *Unity* и многое другое. Это позволяет разработчикам создавать комплексные решения, интегрируя различные компоненты в единую систему.

5 Обширное сообщество и поддержка. *C*# имеет огромное сообщество разработчиков по всему миру, что обеспечивает доступ к множеству ресурсов, статей, форумов и библиотек. Это также означает, что есть множество опытных разработчиков, готовых помочь с возникающими вопросами и проблемами.

6 Поддержка современных тенденций разработки. *C*# постоянно развивается и обновляется, чтобы соответствовать современным требованиям разработки. Новые версии языка и платформы .*NET* внедряют новые возможности, улучшения производительности и инструменты для более эффективного программирования.

*Visual* *Studio* это мощная и универсальная интегрированная среда разработки (*IDE*) от *Microsoft*, предназначенная для создания различных типов приложений, включая веб-приложения, десктопные приложения, мобильные приложения. Интерфейс *Visual* *Studio* 2022 представлен на рисунке 2.2.

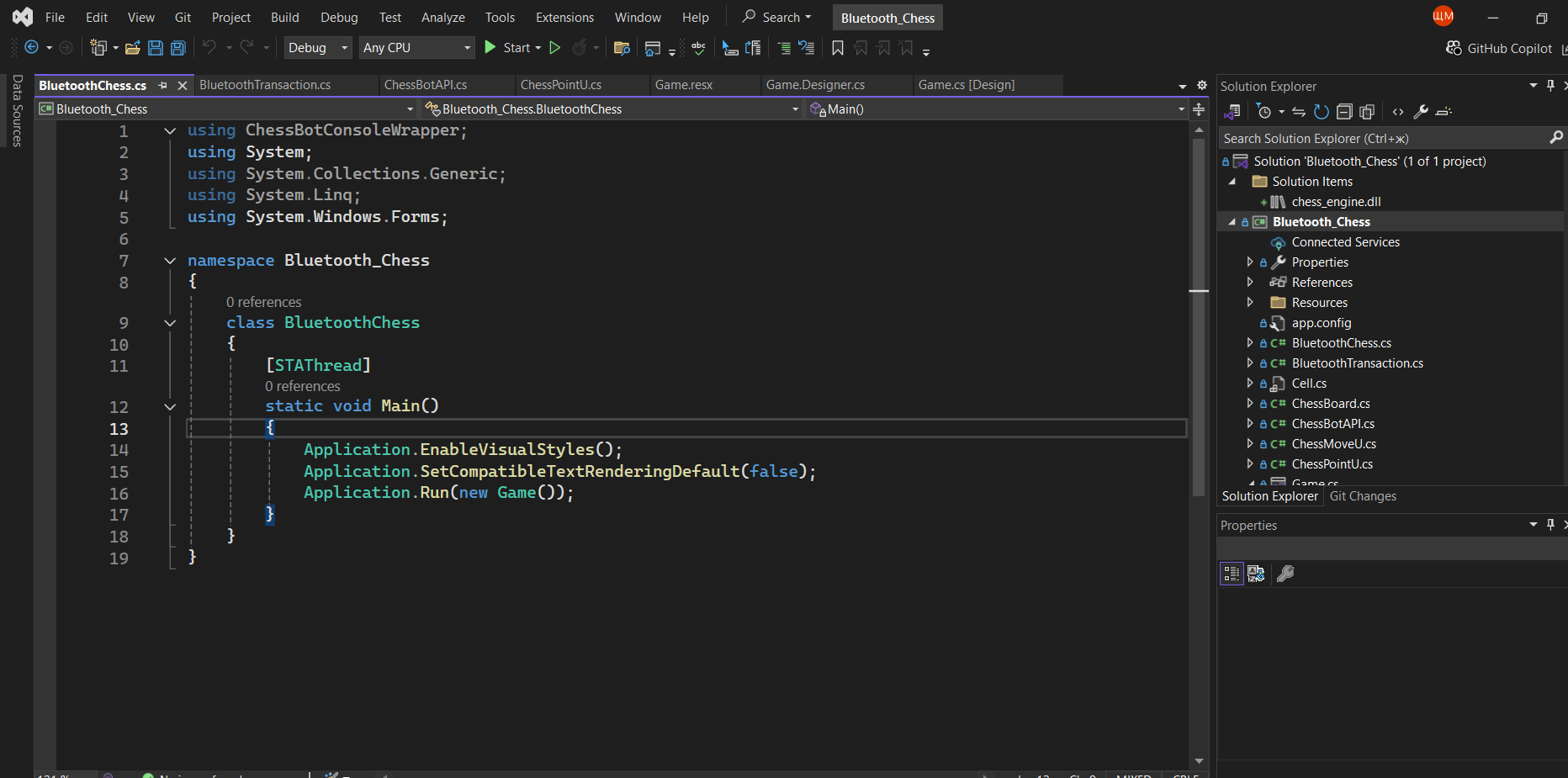


Рисунок 2.2 – Интерфейс *Visual* *Studio* 2022

Эта среда разработки предоставляет разработчикам широкий спектр инструментов и функциональности, что делает процесс создания программного обеспечения более эффективным и продуктивным. В качестве основных особенностей *Visual* *Studio* можно выделить:

1 *Visual* *Studio* объединяет в себе все необходимые инструменты для разработки, включая редактор кода, отладчик, дизайнер интерфейсов, инструменты для сборки и развертывания приложений, а также множество других функций, что позволяет разработчикам работать над проектами в одной удобной среде.

1 *Visual* *Studio* поддерживает не только язык программирования *C*#, но и другие популярные языки, такие как *C*++, *JavaScript*, *Python* и многие другие. Кроме того, она позволяет создавать приложения для различных платформ, включая *Windows*, *Linux*, *macOS*, *Android* и *iOS*.

3 Редактор кода *Visual* *Studio* обладает множеством функций, таких как автозавершение кода, подсветка синтаксиса, быстрые исправления и рефакторинг кода, что упрощает процесс написания и редактирования кода.

4 *Visual* *Studio* предоставляет мощные инструменты для отладки и профилирования приложений, что позволяет быстро выявлять и исправлять ошибки, а также оптимизировать производительность приложений.

В качестве преимуществ *Visual* *Studio* по сравнению с другими средами разработки можно выделить:

1 Интуитивно понятный интерфейс. Интерфейс *Visual* *Studio* разработан с учетом потребностей разработчиков и обеспечивает удобство и интуитивно понятное использование.

2 Мощные инструменты разработки. *Visual* *Studio* предоставляет множество инструментов и функций, которые позволяют разработчикам создавать качественное программное обеспечение быстро и эффективно.

3 Широкие возможности настройки. Среда разработки позволяет настраивать рабочее пространство и функциональность в соответствии с индивидуальными предпочтениями разработчика.

4 Поддержка сообщества и обучение. *Visual* *Studio* имеет активное сообщество разработчиков, где можно найти множество полезных ресурсов, статей, обучающих материалов и форумов для обмена опытом и решения возникающих вопросов.

5 Стабильность и надежность. *Visual* *Studio* предоставляет стабильное и надежное рабочее окружение для разработчиков, что позволяет им сосредоточиться на создании качественного программного обеспечения, не беспокоясь о технических проблемах.

Из-за указанных факторов, а также интеграции *Visual* *Studio* с языком программирования *C*#, данная среда разработки хорошо сочетается с выбранным языком программирования. Их взаимодействие обеспечивает удобство, эффективность и высокую производительность при создании программного обеспечения. Кроме того, широкие возможности настройки *Visual* *Studio* позволяют адаптировать среду разработки под индивидуальные потребности и стиль работы, что способствует повышению производительности и удовлетворенности процессом разработки. Таким образом, было выбрано совместное использование *C*# и *Visual* *Studio*, так как они предоставляют оптимальные условия для создания программного обеспечения.

## **2.3 История, версии и достоинства**

Язык программирования *C*# был разработан командой инженеров в компании *Microsoft* в конце 1990-х годов.

Разработка *C*# началась в 1999 году, а в 2000 году компания *Microsoft* выпустила первую версию языка. Ориентированный на создание программного обеспечения для платформы *Microsoft* .*NET*, *C*# отличается от других языков тем, что был специально разработан для работы в рамках этой платформы.

Ключевыми разработчиками *C*# были Андерс Хейлсберг и его команда. Они разрабатывали язык с учетом опыта, полученного при создании других языков программирования, включая *C*++, *Java* и *Delphi*. Они создали *C*# как часть платформы .*NET* для разработки приложений для операционных систем *Windows*. Однако, *C*# имеет свои особенности, делающие его уникальным.

*C*# сочетает в себе преимущества языков *C*++ и *Java*, такие как управляемый код, сборка мусора и механизмы обработки исключений. Однако, обладает синтаксическими особенностями и расширенными возможностями, которые делают его более удобным для разработки универсальных приложений.

В начале своего развития *C*# был ориентирован на создание *Windows*-приложений, однако со временем он стал использоваться и в других сферах, таких как веб-разработка и разработка мобильных приложений. Основные цели, которые ставили перед собой разработчики *C*#, включали в себя обеспечение простоты использования, надежности и безопасности языка, а также возможность разработки многоплатформенных приложений.

Язык *C*# стал основным языком программирования для платформы .*NET* и получил широкое распространение в различных сферах разработки программного обеспечения. Он используется для создания веб-приложений, настольных приложений, игр, мобильных приложений и многого другого.

Благодаря своим преимуществам, *C*# стал одним из наиболее популярных языков программирования. Он широко используется для разработки приложений на платформе .*NET* и является основным языком программирования для создания приложений для операционной системы *Windows*. С течением времени *C*# продолжает развиваться и обновляться. Компания *Microsoft* регулярно выпускает новые версии языка, добавляя новые функциональные возможности и улучшения. Благодаря этому разработчики могут использовать более современные инструменты и методы программирования.

С течением времени *C*# получал обновления и новые версии, каждая из которых вносила некоторые изменения и новые возможности в язык. Сегодня последней версией является *C*# 12, выпущенная в 2023 году вместе с платформой .*NET* 8.0.

История и развитие языка *C*# говорят о его значимости и значительном вкладе в сферу разработки программного обеспечения. Он продолжает оставаться одним из основных инструментов для создания различных видов приложений.

В заключение, язык программирования *C*# является одним из ключевых инструментов для разработки программного обеспечения в рамках платформы .*NET*, и его история развития свидетельствует о его популярности и значимости в области разработки.

*Visual* *Studio* (*VS*) – это мощная интегрированная среда разработки (*IDE*) для создания разнообразных приложений на платформе *Microsoft*. Разработанная компанией *Microsoft*, *Visual* *Studio* предоставляет широкий спектр инструментов и функциональности для удобной и эффективной работы.

*Visual* *Studio* 1.0 (1997). Первая версия *Visual* *Studio* была выпущена в 1997 году. Она предоставила основные инструменты разработки приложений для *Windows*, такие как редактор кода, отладчик и дизайнер форм.

С течением времени *Visual* *Studio* эволюционировала и стала обладать более широким спектром функциональности, включая интеграцию с различными языками программирования, инструменты для работы с базами данных, средства разработки веб-приложений и мобильных приложений, а также возможности для работы в команде и совместной разработки.

*Visual* *Studio* предоставляет разработчикам удобный и функциональный редактор кода с поддержкой автодополнения, подсветки синтаксиса, интегрированным рефакторингом и другими полезными функциями.

*Visual* *Studio* предоставляет широкий спектр инструментов для отладки приложений, включая точечные остановы, просмотр переменных и стека вызовов, а также возможность отладки в реальном времени.

Таким образом, *Visual* *Studio* является мощной и универсальной средой разработки, которая предоставляет разработчикам все необходимые инструменты для создания качественного программного обеспечения на платформе *Microsoft*.

# **ЗАКЛЮЧЕНИЕ**

# **СПИСОК ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ**

# **ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Листинг программного кода**

# **ПРИЛОЖЕНИЕ Б**

**(обязательное)**

**Функциональная схема алгоритма**

# **ПРИЛОЖЕНИЕ В**

**(обязательное)**

**Блок-схема алгоритма**

# **ПРИЛОЖЕНИЕ Г**

**(обязательное)**

**Графический интерфейс пользователя**

# **ПРИЛОЖЕНИЕ Д**

**(обязательное)**

**Ведомость курсового проекта**