

## ЛР 4. Изучение интегрированных сред разработки

Интегрированная среда разработки (integrated development environment - IDE) – это программное приложение, которое помогает программистам эффективно разрабатывать программный код. Оно повышает производительность разработчиков, объединяя такие возможности, как редактирование, создание, тестирование и упаковка программного обеспечения в простом для использования приложении.

Интегрированная среда разработки (integrated development environment - IDE) - набор инструментов для разработки и отладки программ, имеющий общую интерактивную графическую оболочку, поддерживающую выполнение всех основных функций жизненного цикла разработки программы - набор и редактирование исходного кода, компиляцию (сборку), исполнение, отладку и др.

Более ранний подход системы *UNIX* - использование наборов инструментов (toolkit, toolbox), родственных по тематике и функциональности, но не объединенных в одну интегрированную интерактивную среду и вызываемых в режиме командной строки

Для каждой IDE характерно наличие следующих *компонент*:

- **Единая интерактивная оболочка**, обеспечивающая вызов всех других компонент, не выходя из среды, с широким использованием функциональных клавиш
- **Текстовый редактор** для написания кода
- **Система поддержки сборки (build)**, то есть компиляции проектов из исходных кодов, включающая компилятор с исходного реализуемого языка и **компоновщик** объектных бинарных кодов в единый исполняемый код
- **Отладчик (debugger)** для отладки программ в среде с помощью типичного набора команд: установить контрольную точку остановки; остановиться в заданной процедуре (методе); визуализировать значения переменных

Современные текстовые редакторы в интегрированных средах обеспечивают также режим **автоматического завершения кода**, который в них включен по умолчанию и в котором редактор среды подсказывает разработчику кода возможные и синтаксически правильные его продолжения - например, отсутствие закрывающей скобки, отсутствие точки с запятой; возможные варианты имен методов при вызове метода от объекта какого-либо определенного класса, и т.д.

В современных версиях интегрированных сред появились также следующие возможности:

- **Профилировщик** - инструмент для накопления и анализа статистических данных, полученных в результате исполнения программы под управлением интегрированной среды: число вызовов процедур (методов), объем памяти, используемой при выполнении программы, и т.д.
- **Рефакторинг** - инструментарий **систематических групповых модификаций программ в среде, без принципиальных изменений их функциональности**, с целью улучшения кода. К типичным подобным действиям относится, например, изменение имени метода в его определении и во всех использованиях, добавление его аргумента и т.п.

- **Генератор тестов** - инструмент для генерации типовых тестов для тестирования модулей с различными возможными сочетаниями значений аргументов
- **Система управления версиями исходных кодов** - инструмент интеграции среды с одной из существующих версионных систем
- **Инструменты поддержки командной разработки программ** - этапов жизненного цикла программы распределения заданий по разработке среди участников команды программистов, контроля выполнения заданий менеджером проекта.
- **Инструменты анализа кода** - его семантической корректности: отсутствие некоторых видов ошибок, обнаруживаемых обычно при исполнении, например, недостижимые условия; отсутствие необходимых проверок и полномочий безопасности и др.
- **Инструменты «запутывания» кода**, выполняющие именно с этой целью замену имен элементов кода - классов, методов, полей и т.д. на непонятные, "случайные", "запутанные" имена, с целью затруднения изучения декомпилированного бинарного кода, для защиты от "взлома" кода злоумышленниками, которые хотят несанкционированным образом присвоить себе новые идеи, содержащиеся в коде, либо изучить его со злонамеренными целями организации атак.
- **Поддержка создания различных видов программных проектов и решений** на основе **типовых шаблонов кода (patterns)**. При современной разработке программ подчас требуется создавать очень сильно отличающиеся друг от друга разновидности приложений. Современные интегрированные среды автоматизируют создание различного рода проектов, предоставляя шаблоны исходного кода и генерируя автоматически необходимые для проекта конфигурационные файлы. Трудно представить в настоящее время программирование без использования готовых шаблонов кода, которое неизбежно будет чревато ошибками, - например, очень легко забыть вручную создать тот или иной файл, неотъемлемую часть проекта, либо упустить из вида какой-либо важный фрагмент кода. Поэтому поддержка интегрированными средами различных видов проектов особенно важна. Кроме того, набор возможных видов проектов в современных интегрированных средах является **расширяемым**, т.е. разработчик может ввести в среду при необходимости новый вид проекта.

### Типы IDE:

По компаниям: JB и все другие

По количеству языков: моноязыковые и многоязыковые интегрированные среды (Турбо-Паскаль, *Visual Studio*)

По типу местонахождения: облачные (Colide, Codeanywhere) /десктопные

### Продукты JetBrains:

Licensed to: **Екатерина Жаркова**  
License restriction: For educational use only  
Valid through: February 17, 2023  
Following products included:

- [AppCode](#)
- [CLion](#)
- [DataGrip](#)
- [DataSpell](#)
- [dotCover](#)
- [dotMemory](#)
- [dotTrace](#)
- [GoLand](#)
- [IntelliJ IDEA Ultimate](#)
- [PhpStorm](#)
- [PyCharm](#)
- [ReSharper](#)
- [ReSharper C++](#)
- [Rider](#)
- [RubyMine](#)
- [WebStorm](#)

Первой IDE JetBrains стала IntelliJ IDEA.

WebStorm, PhpStorm, IntelliJ IDEA, PyCharm, Rider, AppCode, CLion, DataGrip, RubyMine, GoLand — все это IDE JetBrains. Их так много, что даже сотрудники JetBrains не всегда точно знают, чем они отличаются.

У всех IDE общее ядро под названием IntelliJ Platform. Все они построены на этой платформе и состоят из различных наборов плагинов. Например:

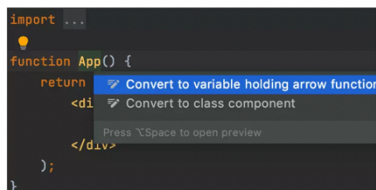
- WebStorm = возможности платформы + плагин для JavaScript и TypeScript + плагин для CSS + около 20 других плагинов для работы с JavaScript.
- PhpStorm = возможности платформы + плагин для JavaScript и TypeScript + плагин для CSS + плагин для PHP + плагин для баз данных + около 30 других плагинов для работы с PHP и JavaScript.

## Продукты Windows:

Visual Studio  
Visual Studio Code

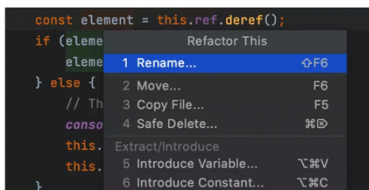
## WebStorm

<https://www.jetbrains.com/ru-ru/webstorm/features/>



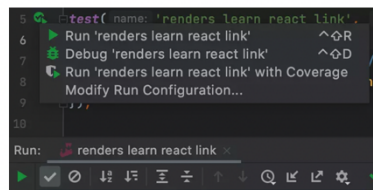
## IDE для JavaScript

Ищете инструмент, которому можно доверить даже самые сложные аспекты JavaScript-разработки? Уже больше 10 лет мы работаем над WebStorm, стремясь максимально упростить разработку на JavaScript. В WebStorm есть все необходимое для работы с JS, TS, React, Vue, Angular, Node.js, HTML, и файлами стилей.



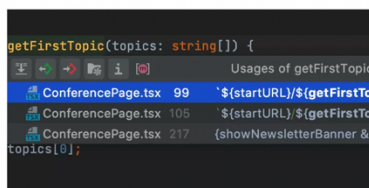
## Умный редактор кода

WebStorm хорошо понимает структуру проектов и поможет вам с любыми аспектами написания кода. Автодополнение кода, безопасный рефакторинг, постоянный поиск потенциальных проблем и подсказки по их исправлению всегда будут у вас под рукой.



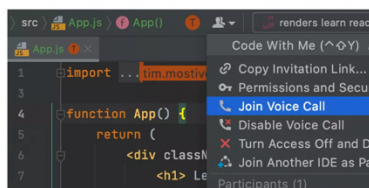
## Встроенные инструменты для разработчиков

Один из главных плюсов IDE в том, что они объединяют все необходимые вам инструменты. Используйте WebStorm для отладки и тестирования клиентского кода и приложений на Node.js, а также для работы с системой контроля версий. Пользуйтесь интегрированными линтерами, инструментами сборки, терминалом и HTTP-клиентом.



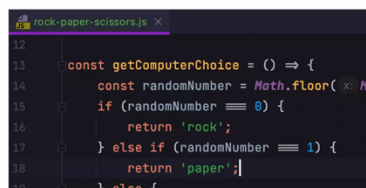
## Быстрая навигация и поиск

Быстро перемещайтесь по коду вне зависимости от размеров проекта. Ищите файлы, классы или символы, просматривая все результаты в одном месте. Переходите к определениям функций, методов, переменных, компонент или классов, находите их использования буквально в пару кликов.



## Эффективная командная работа

Упростите онбординг новых сотрудников – вся функциональность доступна из коробки. Программируйте вместе с командой в реальном времени и общайтесь с коллегами прямо из IDE. Делитесь конфигурацией проекта, в том числе настройками стиля кода. Извлекайте максимум пользы из интеграции с Git и GitHub.



## Кастомизация

Хотите расширить встроенную функциональность? Настройте интерфейс WebStorm так, как нравится вам, при помощи различных тем и плагинов. Вы можете сохранить настройки и использовать их в других инсталляциях WebStorm.

# Основные функциональные возможности

## Плагины



1. Theme plugins - темы
2. IDE Features Trainer – интерактивный учитель
3. Material Theme UI - структурирует все, что можно

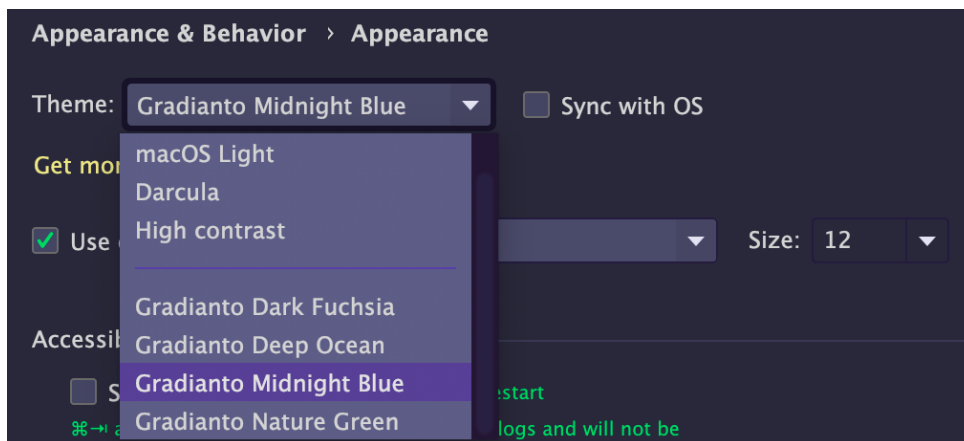
4. **IDEA Mind Map**— интеллект-карта проекта  
<https://habr.com/ru/post/576012/>
5. **Rainbow Brackets** – видна вложенность благодаря выделению цветом
6. Codota - сканирует репозитории на GitHub и анализирует их исходный код. На основе полученных данных плагин предлагает программисту «суперумные» варианты автозавершения кода.
7. Comments Highlighter – раскрашивает комментарии разными цветами в зависимости от их характера и значимости.
8. Return Highlighter - подсвечивает все ключевые слова `return` и помечает строки, в которых они встречаются.
9. **Key Promoter X** - каждый раз, когда вы кликаете мышкой, этот плагин показывает клавиши, которыми можно бы было этот клик заменить. А ещё плагин показывает, какие кнопки интерфейса и как часто вы нажимаете. Это помогает понять, в каких случаях выгоднее запомнить соответствующее сочетание клавиш.
10. **String Manipulation** – редактирование строк. Например, можно преобразовать их к camelCase, kebab-lowercase, KEBAB-UPPERCASE, snake\_case и тп.
11. **EduTools** - позволяет изучать и преподавать языки программирования прямо в IDE. Установив этот плагин, можно пройти не только общедоступный курс по программированию интегрированный в IDE, но и курс, разработанный вашим преподавателем или коллегой. EduTools дает возможность создавать упражнения и делиться ими с коллегами.
12. **GitToolBox** - Webstorm уже поддерживает полнофункциональную интеграцию с Git, однако этот плагин открывает новые возможности, которые можно настроить под себя. В основном его используют из-за удобной функции просмотра истории строки: с помощью аннотации можно легко узнать, когда и кем редактировалась та или иная строка.
13. **WakaTime** – это как фитнес-трекер, только служит он для мониторинга вашей активности при написании кода. Плагин *WakaTime* контролирует время, ведет тщательный учет и составляет точную аналитическую картину вашей деятельности, а также автоматически генерирует статистические отчеты. Его можно использовать как для оценки эффективности работы команды, так и для поиска способов оптимизации собственной продуктивности.  
<https://wakatime.com/settings/api-key>
14. Database Tools and SQL - для работы с базами данных (но только по лицензии JB)

Установка плагина с диска:

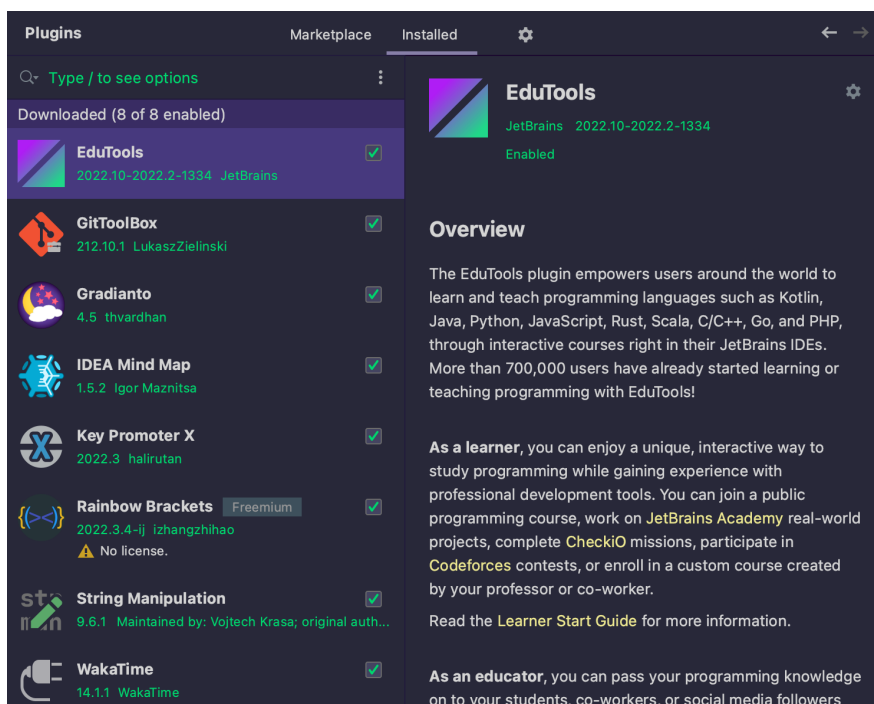
#### Gradio Plugin Is Being Downloaded

##### How to Install 'Gradio'

1. Install an IDE if you don't have one:
2. Open your IDE and press  to open the IDE settings.
3. Select **Plugins**, click  and then click **Install Plugin from Disk**.
4. Select the plugin archive file and click **OK**.
5. Click **OK** to apply the changes and restart your IDE if prompted.



А вообще устанавливаются прямо с IDE:



## Интеграция с VCS

<https://www.jetbrains.com/help/webstorm/version-control-integration.html>

<https://javarush.com/groups/posts/2818-podruzhim-git-s-intellij-idea>

## Интеграция с системами сборки

Maven, Gradle, Ant, Gant

<https://www.jetbrains.com/help/idea/compiling-applications.html>

<https://www.jetbrains.com/ru-ru/teamcity/tutorials/dotnet-build-configure-test/> - TeamCity

Системы сборки могут управлять процессом сборки ваших проектов. Этот инструмент не зависит от IDE. Он позволяет другим людям легко собирать ваш плагин из исходных файлов.

## Список горячих клавиш IDE:

<https://blog.listratenkov.com/webstorm-ide-hot-keys/>

### Навигация по проекту:

Double Shift	Поиск повсюду
Command + E	Выбрать последние файлы
Ctrl + Option + ➡/⬅	Изменить размер вертикального tool window
Ctrl + Option + ⬇/⬆	Изменить размер горизонтального tool window
Command + W	Закрыть активную вкладку
Shift + Command + [ / ]	Перейти к предыдущей/следующей вкладке
Ctrl + Tab	Переключение между вкладками инструментами
Command + [ / ]	Вперед/Назад
Command + Shift + V	Открыть буфер

### Навигация по файлу:

Command + Option + Shift + [ / ]	Выбрать весь код вверх/вниз
Command + -/+	Свернуть/развернуть блок кода
Ctrl + M	Перейти к парной скобке
Option + ➡/⬅	Хождение по словам
Option + Shift + ➡/⬅	Выделение по словам
Command + F/R	Поиск/Замена
Ctrl + Shift + B	Перейти к объявлению типа
Command + Shift + F7	Выделить случаи использования

### Редактирование файла:

Command + Option + L	Форматирование кода
Command + /	Добавить/удалить комментарии
Command + J	Вставить шаблон из списка
Option + Shift + ⬇/⬆	Переместить строку вниз/вверх
Command + D	Продублировать строку
Command + Delete	Удалить строку
Option + Click	Добавить курсор

### Рефакторинг:

Ctrl + T	Рефакторинг
Command + N	Генерация геттеров/сеттеров

Command + Option + M	Перенести в метод
Command + Option + V	Перенести в переменную
Command + Option + F	Перенести в свойство класса
Command + Option + C	Перенести в константу

### Работа с VCS:

Ctrl + V	Всплывающее окно системы контроля версий
Option + Shift + C	Просмотр последних изменений
Command + K	Коммит
Ctrl + Shift + K	Пуш
Command + T	Пул
Command + 9	Открыть/закрыть гит
Command + 0	Открыть/закрыть вкладку комитов

### Запуск/Отладка/Тестирование:

Ctrl + R	Выполнить
Ctrl + D	Отладить
Ctrl + Option + R	Выбрать конфигурацию и выполнить
Ctrl + Option + D	Выбрать конфигурацию и отладить
Ctrl + Command + R	Запуск тестов
Option + F9	Отладка от курсора
Command + F8	Установить/снять контрольную точку
F8/F7	Шаг с обходом/заходом

*шаг с заходом* будет входить в вызовы функций и выполнять их построчно, *шаг с обходом* выполнит всю функцию без остановки и вернет вам управление после выполнения этой функции.