# Документация

на Велин Яворски 12А №5

на тема: Програма за филтриране на изображения

# Увод

Приложението предоставя на потребителите лесен интерфейс, с който да заредят свое изображение в програмата, и да му приложат един или няколко филтъра по свой избор с цел постигане на различни ефекти.

# Първа глава Подбор на езика/развойната среда

* 1. Подбор на език

За разработката на приложението е използван езикът **Java**. Той е език от високо ниво, което означава, че върши повече неща с по-малко команди, по-лесен е за писане и е по-лесен за използване. В Java има и вградени библиотеки за обработка на изображения като например Graphics2D, BufferedImage, Kernel, ConvoleOp, с които лесно може да се имплементират различни филтри за изображения.

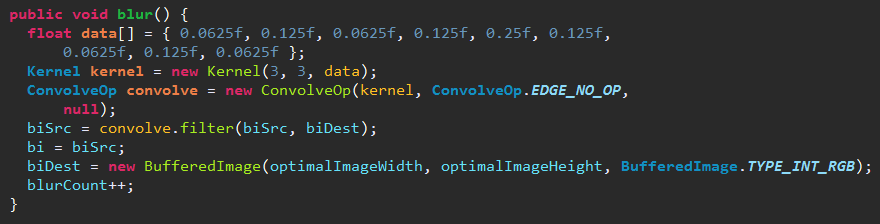
* 1. Подбор на развойна среда

За развойна среда е използван **Eclipse**, който представлява многоезична среда за разработване на софтуер, която включва интегрирана среда за разработка (IDE) и плъгин система. Благодарение на eclipse software development kit (SDK), който включва инструменти за разработка на Java, необходими на Java разработчиците, и плъгин системата, чрез която разработчиците могат да персонализират своето IDE, писането на Java в Eclipse e много по-лесно в сравнение с повечето платформи за разрабока.

# Втора глава Алгоритъм на приложенито

За момента в приложението са имплементирани три филтъра – blur, sharpen и

edge detect. Всеки филтър е отделна функция в програмата, като с натискане на един от бутоните в програмата се извиква съответната функция. Първоначално в програмата не е заредено изображение и функциите не са активни. Когато потребителят зареди някакво изображение се преинициализира създадената инстанция на класа CPanel, който представлява панелa, в който изображението се рендерира и, в който се съдържат всички функции нужни, за да се извърши филтрирането на изображението. Преинициализирането се състои в зареждане на изображението в програмата, което става във функцията loadImage(), инициализирането на променливите, в които ще се пази филтрираното изображение, инициализирането на графичния обект, чрез който ще се рендерира филтрираното изображение и рендерирането на първоначално зареденото изображение. Последните няколко инициализации стават във фукцията createBufferedImages().

Функциите blur(), sharpen(), и edgeDetect() прилагат съответните филтри на зареденото изображение. Те са идентични с изключение на масива data, в който се пазят стойностите, с които ще бъде съставена матрицата, с която изображението ще бъде обработено. Променливата kernel представлява самата матрица. Чрез променливата convolve се извършва прилагането на матрицата към изображението. В променливата biSrc се запазва новото филтрирано изображение след изпълнението на функцията filter(). Функцията filter() приема 2 аргумента, първият е изображението, което ще бъде филтрирано, а вторият, променлива от тип BufferedImage, в която ще се запази новото изображение. Понеже обаче функцията връща като резултат филтрираното изображение е по-удобно то да се запази отново в променливата biSrc вместо да се използва biDest, защото е по-лесно да се имплементира наслагването на филтри върху изображението. След всяко изпълнение на една от тези функции панелът, в който се рендерира изображението се пребоядисва. Променливата bi се използва, за да се посочи кое изображение да се рендерира при пребоядисването му.

Алгоритъм за прилагане на blur филтър