Отчет по индивидуальному заданию

Вариант №9 «Конвертер BMP»

по дисциплине  
«Архитектура ЭВМ»

|  |  |
| --- | --- |
| Выполнил  студент гр. 3530904/80004 | Селезнев В. А. |
| Преподаватель | Петров А. В. |
|  |  |

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2020 г.

**Оглавление**

[Задание 3](#_Toc39665346)

[Код программы 4](#_Toc39665347)

[Результаты работы программы 20](#_Toc39665348)

# **Задание**

Конвертор BMP в BMP с графическим интерфейсом. Программа по выбору конвертирует изображения из 24 бит на пиксель в 1, 4, 8, 16, 24 бит на пиксель.

# **Код программы**

Основная часть программы написана на языке C++:

Файл 16Bit.h

#include <windows.h>

struct Pixel16

{

BYTE first;

BYTE second;

};

void pixelConversionTo16Bit(int& width, int& height, HANDLE& hInputFile, HANDLE& hOutFile,

Pixel16\* outBuf, RGBTRIPLE\* inBuf);

Файл 16Bit.cpp

#include "16Bit.h"

#include "getOutBufSize.h"

#include "padding.h"

#include <math.h>

void pixelConversionTo16Bit(int& width, int& height, HANDLE& hInputFile, HANDLE& hOutFile,

Pixel16\* outBuf, RGBTRIPLE\* inBuf)

{

DWORD RW;

unsigned char bit = 16;

for (int i = 0; i < height; ++i)

{

ReadFile(hInputFile, inBuf, sizeof(RGBTRIPLE) \* width, &RW, NULL);

for (int j = 0; j < width; ++j)

{

auto red = (BYTE)round(inBuf[j].rgbtRed \* (double)31 / (double)255);

auto green = (BYTE)round(inBuf[j].rgbtGreen \* (double)31 / (double)255);

auto blue = (BYTE)round(inBuf[j].rgbtBlue \* (double)31 / (double)255);

auto greenFirst = (green & 0x07) << 5;

auto greenSecond = (green >> 3);

outBuf[j].first = blue;

outBuf[j].first |= greenFirst;

outBuf[j].second = red;

outBuf[j].second = (outBuf[j].second << 2);

outBuf[j].second |= greenSecond;

}

WriteFile(hOutFile, outBuf, sizeof(Pixel16) \* getOutBufSize(width, bit), &RW, NULL);

addPadding(width, hOutFile, bit);

}

}

Файл 8Bit.h

#include <windows.h>

void getPaletteFor8Bit(RGBQUAD\* Palette);

void pixelConversionTo8Bit(int& width, int& height, HANDLE& hInputFile, HANDLE& hOutFile,

BYTE\* outBuf, RGBTRIPLE\* inBuf, RGBQUAD Palette[]);

Файл 8Bit.cpp

#include "8Bit.h"

#include "getOutBufSize.h"

#include "padding.h"

void getPaletteFor8Bit(RGBQUAD\* Palette)

{

for (int i = 0; i < 8; ++i)

{

for (int j = 0; j < 8; ++j)

{

for (int k = 0; k < 4; ++k)

{

Palette[i \* 32 + j \* 4 + k].rgbRed = i \* 32;

Palette[i \* 32 + j \* 4 + k].rgbGreen = j \* 32;

Palette[i \* 32 + j \* 4 + k].rgbBlue = k \* 64;

}

}

}

}

void pixelConversionTo8Bit(int& width, int& height, HANDLE& hInputFile, HANDLE& hOutFile,

BYTE\* outBuf, RGBTRIPLE\* inBuf, RGBQUAD Palette[])

{

DWORD RW;

unsigned char bit = 8;

for (int i = 0; i < height; ++i)

{

ReadFile(hInputFile, inBuf, sizeof(RGBTRIPLE) \* width, &RW, NULL);

for (int j = 0; j < width; ++j)

{

outBuf[j] = 32 \* (inBuf[j].rgbtRed / 32) + 4 \* (inBuf[j].rgbtGreen / 32) + inBuf[j].rgbtBlue / 64;

}

WriteFile(hOutFile, outBuf, sizeof(BYTE) \* getOutBufSize(width, bit), &RW, NULL);

addPadding(width, hOutFile, bit);

}

}

Файл 4Bit.h

#include <windows.h>

void getPaletteFor4Bit(RGBQUAD\* Palette);

void pixelConversionTo4Bit(int& width, int& height, HANDLE& hInputFile, HANDLE& hOutFile,

BYTE\* outBuf, RGBTRIPLE\* inBuf, RGBQUAD Palette[]);

Файл 4Bit.cpp

#include "4Bit.h"

#include "getOutBufSize.h"

#include "padding.h"

#include <math.h>

void getPaletteFor4Bit(RGBQUAD\* Palette)

{

Palette[0].rgbRed = 0x00;

Palette[0].rgbGreen = 0x00;

Palette[0].rgbBlue = 0x00;

Palette[1].rgbRed = 128;

Palette[1].rgbGreen = 0;

Palette[1].rgbBlue = 0;

Palette[2].rgbRed = 255;

Palette[2].rgbGreen = 0;

Palette[2].rgbBlue = 0;

Palette[3].rgbRed = 255;

Palette[3].rgbGreen = 0;

Palette[3].rgbBlue = 255;

Palette[4].rgbRed = 0;

Palette[4].rgbGreen = 128;

Palette[4].rgbBlue = 128;

Palette[5].rgbRed = 0;

Palette[5].rgbGreen = 128;

Palette[5].rgbBlue = 0;

Palette[6].rgbRed = 0;

Palette[6].rgbGreen = 255;

Palette[6].rgbBlue = 0;

Palette[7].rgbRed = 0;

Palette[7].rgbGreen = 255;

Palette[7].rgbBlue = 255;

Palette[8].rgbRed = 0;

Palette[8].rgbGreen = 0;

Palette[8].rgbBlue = 128;

Palette[9].rgbRed = 128;

Palette[9].rgbGreen = 0;

Palette[9].rgbBlue = 128;

Palette[10].rgbRed = 0;

Palette[10].rgbGreen = 0;

Palette[10].rgbBlue = 255;

Palette[11].rgbRed = 192;

Palette[11].rgbGreen = 192;

Palette[11].rgbBlue = 192;

Palette[12].rgbRed = 128;

Palette[12].rgbGreen = 128;

Palette[12].rgbBlue = 128;

Palette[13].rgbRed = 128;

Palette[13].rgbGreen = 128;

Palette[13].rgbBlue = 0;

Palette[14].rgbRed = 255;

Palette[14].rgbGreen = 255;

Palette[14].rgbBlue = 0;

Palette[15].rgbRed = 255;

Palette[15].rgbGreen = 255;

Palette[15].rgbBlue = 255;

}

void pixelConversionTo4Bit(int& width, int& height, HANDLE& hInputFile, HANDLE& hOutFile,

BYTE\* outBuf, RGBTRIPLE\* inBuf, RGBQUAD Palette[])

{

DWORD RW;

unsigned char bit = 4;

for (int i = 0; i < height; ++i)

{

ReadFile(hInputFile, inBuf, sizeof(RGBTRIPLE) \* width, &RW, NULL);

auto pixelMask = new char[2];

int maskIndex = 0;

int outIndex = 0;

for (int j = 0; j < width; ++j)

{

auto sum = inBuf[j].rgbtRed + inBuf[j].rgbtGreen + inBuf[j].rgbtBlue;

pixelMask[maskIndex] = (int)floor(sum \* 15 / (3 \* 255));

maskIndex++;

if (maskIndex == 2)

{

BYTE resultByte = 0;

for (auto k = 0; k < 2; k++)

{

if (k != 0)

resultByte |= pixelMask[k];

else

resultByte |= (pixelMask[k] << 4);

}

outBuf[outIndex] = resultByte;

outIndex++;

maskIndex = 0;

}

}

if (maskIndex != 0)

{

BYTE resultByte = 0;

for (auto k = 0; k < maskIndex; k++)

{

if (k != 0)

resultByte |= pixelMask[k];

else

resultByte |= (pixelMask[k] << 4);

}

outBuf[outIndex] = (resultByte << 2 - maskIndex);

maskIndex = 0;

}

delete[] pixelMask;

WriteFile(hOutFile, outBuf, sizeof(BYTE) \* getOutBufSize(width, bit), &RW, NULL);

addPadding(width, hOutFile, bit);

}

}

Файл 1Bit.h

#include <windows.h>

void getPaletteFor1Bit(RGBQUAD\* Palette);

void pixelConversionTo1Bit(int& width, int& height, HANDLE& hInputFile, HANDLE& hOutFile,

BYTE\* outBuf, RGBTRIPLE\* inBuf, RGBQUAD Palette[]);

Файл 1Bit.cpp

#include "1Bit.h"

#include "getOutBufSize.h"

#include "padding.h"

void getPaletteFor1Bit(RGBQUAD\* Palette)

{

Palette[0].rgbRed = 0x00;

Palette[0].rgbGreen = 0x00;

Palette[0].rgbBlue = 0x00;

Palette[1].rgbRed = 0xFF;

Palette[1].rgbGreen = 0xFF;

Palette[1].rgbBlue = 0xFF;

}

void pixelConversionTo1Bit(int& width, int& height, HANDLE& hInputFile, HANDLE& hOutFile,

BYTE\* outBuf, RGBTRIPLE\* inBuf, RGBQUAD Palette[])

{

DWORD RW;

unsigned char bit = 1;

for (int i = 0; i < height; ++i)

{

ReadFile(hInputFile, inBuf, sizeof(RGBTRIPLE) \* width, &RW, NULL);

auto pixelMask = new char[8];

int maskIndex = 0;

int outIndex = 0;

for (int j = 0; j < width; ++j)

{

auto average = (inBuf[j].rgbtRed + inBuf[j].rgbtGreen + inBuf[j].rgbtBlue) / 3;

pixelMask[maskIndex] = average > 128 ? 1 : 0;

maskIndex++;

if (maskIndex == 8)

{

BYTE resultByte = 0;

for (auto k = 0; k < 8; k++)

{

if (k != 0)

resultByte = resultByte << 1;

resultByte |= (pixelMask[k] == 1 ? 0x01 : 0x00);

}

outBuf[outIndex] = resultByte;

outIndex++;

maskIndex = 0;

}

}

if (maskIndex != 0)

{

BYTE resultByte = 0;

for (auto k = 0; k < maskIndex; k++)

{

if (k != 0)

resultByte = resultByte << 1;

resultByte |= (pixelMask[k] == 1 ? 0x01 : 0x00);

}

outBuf[outIndex] = (resultByte << 8 - maskIndex);

maskIndex = 0;

}

delete[] pixelMask;

WriteFile(hOutFile, outBuf, sizeof(BYTE) \* getOutBufSize(width, bit), &RW, NULL);

addPadding(width, hOutFile, bit);

}

}

Файл getOutBufSize.h

int getOutBufSize(int& width, unsigned char& bit);

Файл getOutBufSize.cpp

#include <math.h>

#include <stdexcept>

#include "getOutBufSize.h"

int getOutBufSize(int& width, unsigned char& bit)

{

if (bit == 1)

{

return (int)ceil(width / (double)8);

}

else if (bit == 4)

{

return (int)ceil(width / (double)2);

}

else if (bit == 8)

{

return width;

}

else if (bit == 16)

{

return width;

}

else

{

throw new std::invalid\_argument("Error");

}

}

Файл padding.h

int getPadding(int& width, unsigned char& bit);

void addPadding(int& width, HANDLE& hOutFile, unsigned char& bit);

Файл padding.cpp

#include <windows.h>

#include "padding.h"

#include "getOutBufSize.h"

int getPadding(int& width, unsigned char& bit)

{

auto pad = getOutBufSize(width, bit) % 4;

if (pad == 0)

return 0;

return 4 - pad;

}

void addPadding(int& width, HANDLE& hOutFile, unsigned char& bit)

{

DWORD RW;

auto paddingSize = getPadding(width, bit);

for (auto t = 0; t < paddingSize; t++)

{

BYTE\* padByte = new BYTE;

\*padByte = 0;

WriteFile(hOutFile, padByte, 1, &RW, NULL);

}

}

Файл main.cpp

#include "convert.h"

int main(int argc, char\* argv[])

{

std::string fileName = argv[1];

unsigned char bit = std::stoi(argv[2]);

convert(fileName, bit);

return 0;

}

Графический интерфейс выполнен на языке Java:

Файл UserInterface.java

package coursework;  
  
import javax.imageio.ImageIO;  
import javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.image.BufferedImage;  
import java.io.File;  
  
public class UserInterface  
{  
 public UserInterface()  
 {  
 JFrame frame = new JFrame();  
 frame.add(rootPanel);  
 frame.setTitle("BMP converter");  
 frame.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame.setSize(900, 700);  
 frame.setLocation(290, 50);  
 frame.setVisible(true);  
  
 comboBit.addItem("24-bit");  
 comboBit.addItem("16-bit");  
 comboBit.addItem("8-bit");  
 comboBit.addItem("4-bit");  
 comboBit.addItem("1-bit");  
 comboBit.setSelectedItem("24-bit");  
 }  
  
 public void interfaceImplementation()  
 {  
 pictureLabel.setText("Here will be your picture");  
 String pathToExe = "C:\\University\\Repositories\\CompArch\\coursework\\Debug\\coursework.exe";  
 String pathToResultImage = "C:\\University\\Repositories\\CompArch\\GUI\\Result.bmp";  
  
 File file = new File(pathToResultImage);  
 file.delete();  
  
 convertButton.addActionListener(new ActionListener()  
 {  
 @Override  
 public void actionPerformed (ActionEvent e)  
 {  
 String bit;  
 String commandValue = (String) comboBit.getSelectedItem();  
 String pathToImage = pathField.getText();  
 pathToImage = pathToImage.trim();  
 errorLabel.setText("");  
  
 try  
 {  
 switch(commandValue)  
 {  
 case "24-bit":;  
 BufferedImage myPicture24 = ImageIO.*read*(new File(pathToImage));  
 ImageIcon image24 = new ImageIcon(myPicture24);  
 pictureLabel.setText("");  
 pictureLabel.setIcon(image24);  
 break;  
  
 case "16-bit":  
 bit = "16";  
 Process process16 = new ProcessBuilder(pathToExe, pathToImage, bit).start();  
 process16.waitFor();  
 BufferedImage myPicture16 = ImageIO.*read*(new File(pathToResultImage));  
 ImageIcon image16 = new ImageIcon(myPicture16);  
 pictureLabel.setText("");  
 pictureLabel.setIcon(image16);  
 break;  
  
 case "8-bit":  
 bit = "8";  
 Process process8 = new ProcessBuilder(pathToExe, pathToImage, bit).start();  
 process8.waitFor();  
 BufferedImage myPicture8 = ImageIO.*read*(new File(pathToResultImage));  
 ImageIcon image8 = new ImageIcon(myPicture8);  
 pictureLabel.setText("");  
 pictureLabel.setIcon(image8);  
 break;  
  
 case "4-bit":  
 bit = "4";  
 Process process4 = new ProcessBuilder(pathToExe, pathToImage, bit).start();  
 process4.waitFor();  
 BufferedImage myPicture4 = ImageIO.*read*(new File(pathToResultImage));  
 ImageIcon image4 = new ImageIcon(myPicture4);  
 pictureLabel.setText("");  
 pictureLabel.setIcon(image4);  
 break;  
  
 case "1-bit":  
 bit = "1";  
 Process process1 = new ProcessBuilder(pathToExe, pathToImage, bit).start();  
 process1.waitFor();  
 BufferedImage myPicture1 = ImageIO.*read*(new File(pathToResultImage));  
 ImageIcon image1 = new ImageIcon(myPicture1);  
 pictureLabel.setText("");  
 pictureLabel.setIcon(image1);  
 break;  
 }  
 }  
 catch(Exception err)  
 {  
 errorLabel.setText("Error");  
 }  
 }  
 });  
 }  
  
 private JPanel rootPanel;  
 private JButton convertButton;  
 private JTextField pathField;  
 private JComboBox<String> comboBit;  
 private JLabel pictureLabel;  
 private JLabel errorLabel;  
}

Файл UserInterface.form

<?xml version="1.0" encoding="UTF-8"?>

<form xmlns="http://www.intellij.com/uidesigner/form/" version="1" bind-to-class="coursework.UserInterface">

<grid id="27dc6" binding="rootPanel" layout-manager="GridLayoutManager" row-count="2" column-count="1" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<xy x="20" y="20" width="500" height="400"/>

</constraints>

<properties/>

<border type="none"/>

<children>

<grid id="9063c" layout-manager="GridLayoutManager" row-count="1" column-count="2" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<grid row="0" column="0" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill="3" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

<border type="none"/>

<children>

<grid id="4cfe1" layout-manager="GridLayoutManager" row-count="3" column-count="1" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<grid row="0" column="0" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill="3" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

<border type="none"/>

<children>

<grid id="c664e" layout-manager="GridLayoutManager" row-count="1" column-count="1" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<grid row="0" column="0" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill="3" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

<border type="none"/>

<children>

<component id="4218f" class="javax.swing.JLabel">

<constraints>

<grid row="0" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="0" anchor="0" fill="0" indent="0" use-parent-layout="false"/>

</constraints>

<properties>

<text value="Write image path:"/>

</properties>

</component>

</children>

</grid>

<grid id="61f2f" layout-manager="GridLayoutManager" row-count="1" column-count="1" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<grid row="1" column="0" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill="3" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

<border type="none"/>

<children>

<component id="f11a0" class="javax.swing.JLabel">

<constraints>

<grid row="0" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="0" anchor="0" fill="0" indent="0" use-parent-layout="false"/>

</constraints>

<properties>

<text value="Select bit:"/>

</properties>

</component>

</children>

</grid>

<grid id="5685b" layout-manager="GridLayoutManager" row-count="1" column-count="1" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<grid row="2" column="0" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill="3" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

<border type="none"/>

<children>

<component id="50c5f" class="javax.swing.JLabel">

<constraints>

<grid row="0" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="0" anchor="0" fill="0" indent="0" use-parent-layout="false"/>

</constraints>

<properties>

<text value="Push the button!"/>

</properties>

</component>

</children>

</grid>

</children>

</grid>

<grid id="be9f0" layout-manager="GridLayoutManager" row-count="3" column-count="1" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<grid row="0" column="1" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill="3" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

<border type="none"/>

<children>

<grid id="fe3f" layout-manager="GridLayoutManager" row-count="1" column-count="3" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<grid row="0" column="0" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill="3" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

<border type="none"/>

<children>

<component id="a506c" class="javax.swing.JTextField" binding="pathField">

<constraints>

<grid row="0" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="6" anchor="0" fill="0" indent="0" use-parent-layout="false">

<preferred-size width="150" height="-1"/>

</grid>

</constraints>

<properties/>

</component>

<grid id="cac0e" layout-manager="GridLayoutManager" row-count="1" column-count="1" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<grid row="0" column="2" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill="3" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

<border type="none"/>

<children/>

</grid>

<component id="7017c" class="javax.swing.JLabel" binding="errorLabel">

<constraints>

<grid row="0" column="1" row-span="1" col-span="1" vsize-policy="0" hsize-policy="0" anchor="8" fill="0" indent="0" use-parent-layout="false"/>

</constraints>

<properties>

<text value=""/>

</properties>

</component>

</children>

</grid>

<grid id="674ad" layout-manager="GridLayoutManager" row-count="1" column-count="1" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<grid row="1" column="0" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill="3" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

<border type="none"/>

<children>

<component id="87d31" class="javax.swing.JComboBox" binding="comboBit">

<constraints>

<grid row="0" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="2" anchor="0" fill="0" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

</component>

</children>

</grid>

<grid id="4da54" layout-manager="GridLayoutManager" row-count="1" column-count="1" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<grid row="2" column="0" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill="3" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

<border type="none"/>

<children>

<component id="b216d" class="javax.swing.JButton" binding="convertButton">

<constraints>

<grid row="0" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="3" anchor="0" fill="0" indent="0" use-parent-layout="false"/>

</constraints>

<properties>

<text value="Convert"/>

</properties>

</component>

</children>

</grid>

</children>

</grid>

</children>

</grid>

<grid id="faa7c" layout-manager="GridLayoutManager" row-count="1" column-count="1" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">

<margin top="0" left="0" bottom="0" right="0"/>

<constraints>

<grid row="1" column="0" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill="3" indent="0" use-parent-layout="false"/>

</constraints>

<properties/>

<border type="none"/>

<children>

<component id="9bfec" class="javax.swing.JLabel" binding="pictureLabel">

<constraints>

<grid row="0" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="0" anchor="0" fill="0" indent="0" use-parent-layout="false"/>

</constraints>

<properties>

<text value=""/>

</properties>

</component>

</children>

</grid>

</children>

</grid>

</form>

Файл Main.java

package coursework;  
  
public class Main  
{  
 public static void main(String[] args)  
 {  
 UserInterface ui = new UserInterface();  
 ui.interfaceImplementation();  
 }  
}

# **Результаты работы программы**

Конвертировать будем следующее 24-битное изображение:



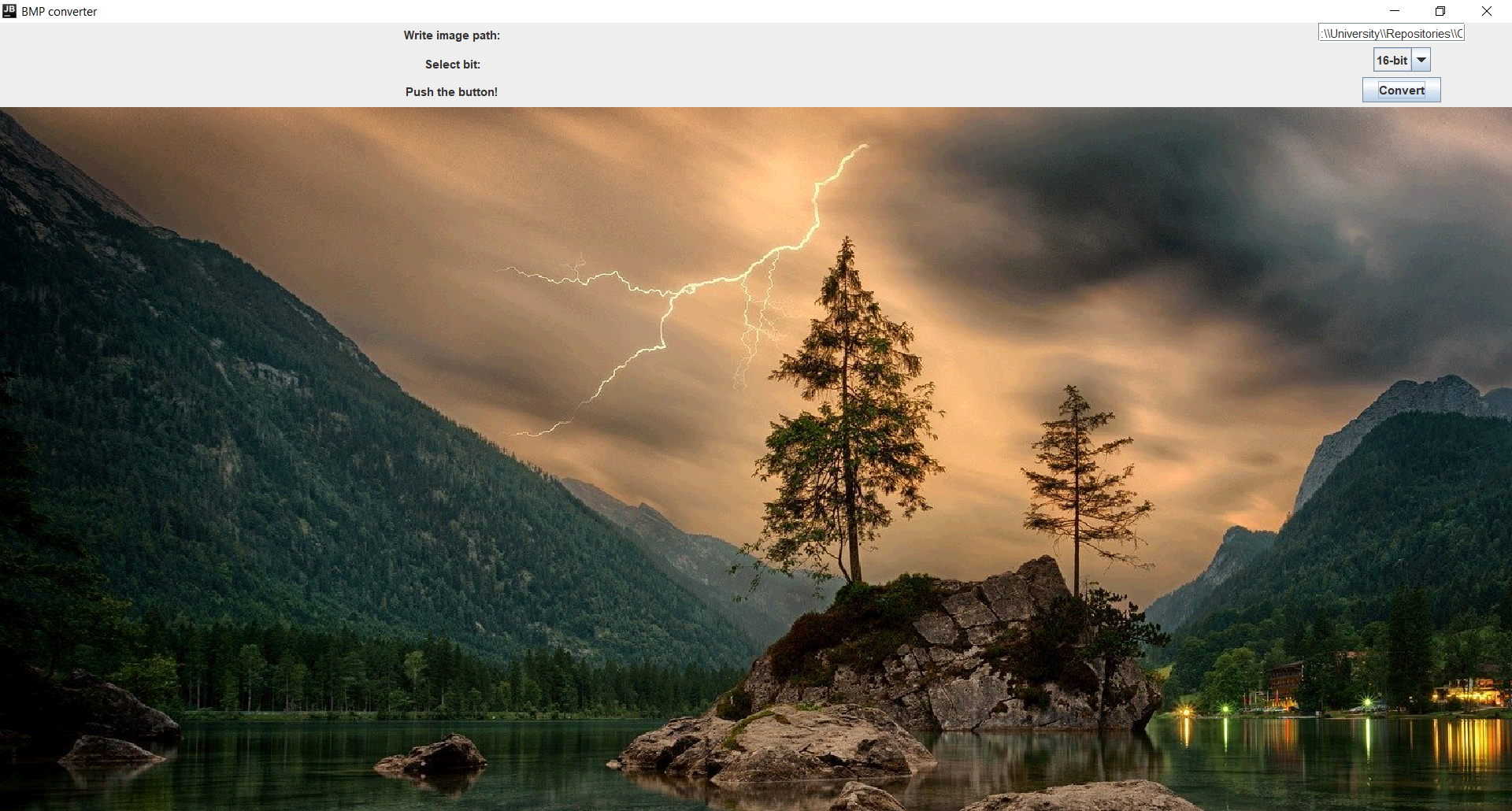
Программа выглядит следующим образом:



Загружаем изображение:



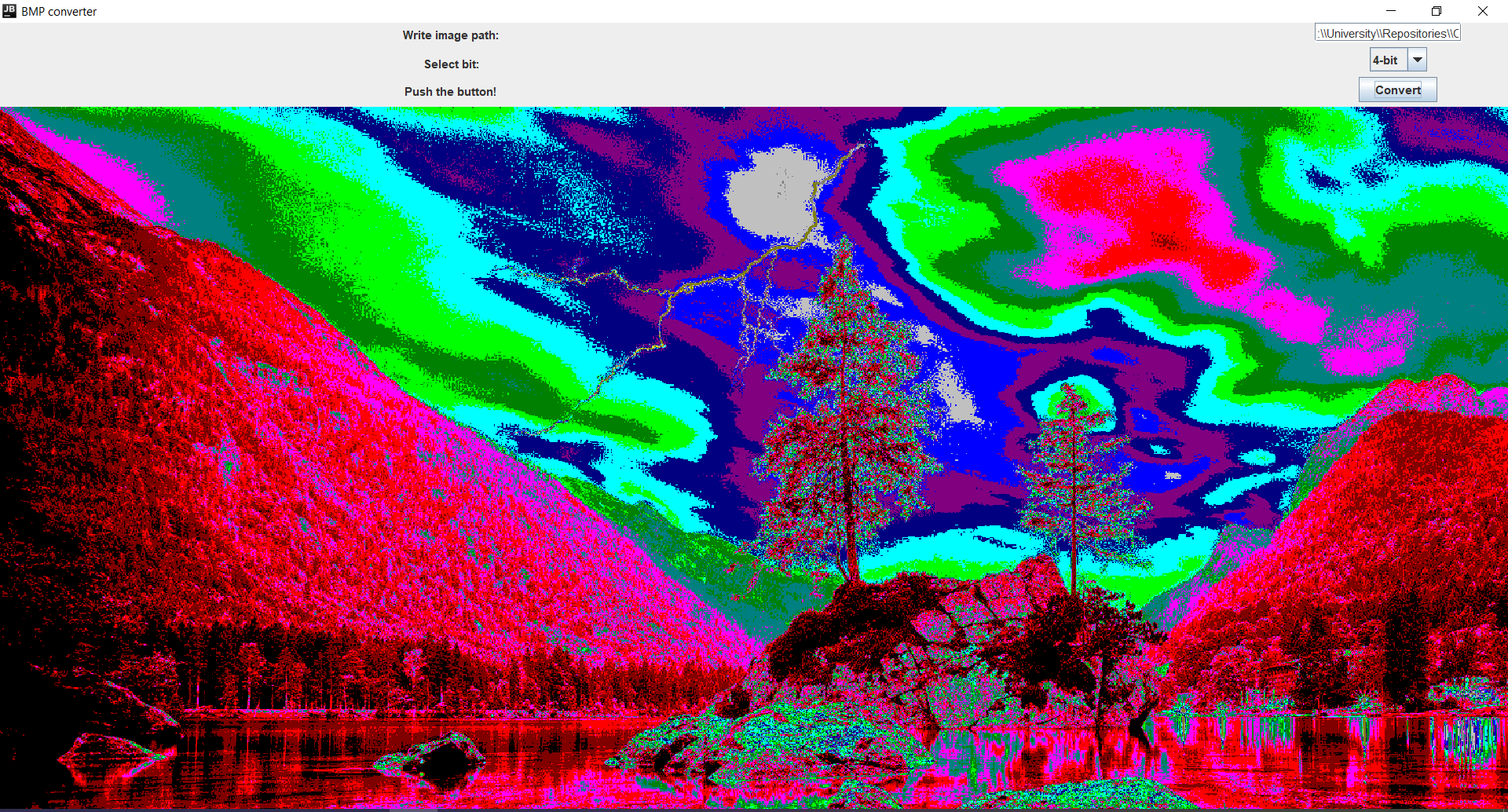
Конвертация в 16-бит:



Конвертация в 8-бит:



Конвертация в 4-бита:



Конвертация в 1-бит:

