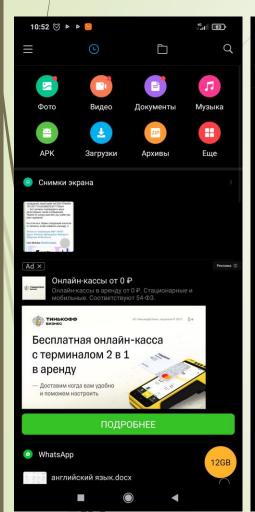
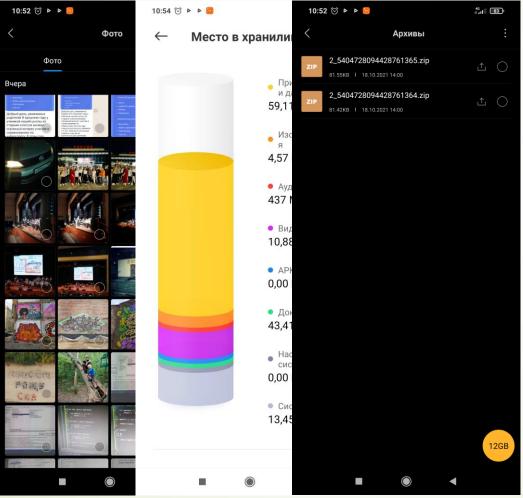
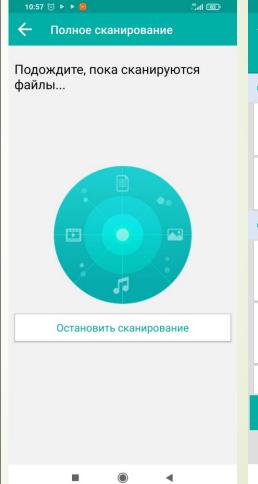
# DuplicateFinder

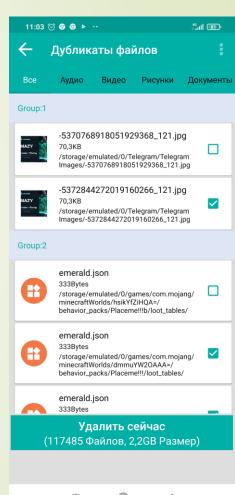
Petr Goncharov

## Introduction









## Target and task

Цель: Сделать удобнее работу с файлами на персональном компьютере

**Задача:** Разработать приложение анализа файлов на дисках персонального компьютера для работы в операционных системах Windows и Linux, которое:

- 1. Отображает структуру расположения файлов и директорий на дисках;
- 2. Классифицирует файлы по типам (видео, аудио, текст, exe-), размеру или дате создания и внесения изменений;
- 3. Находит все дубликаты файлов, сравнивая «побитово» содержимое файлов «примерно» одинакового размера (отображает в процентах степень похожести);
- 4. Индексирует содержимое файла на основе Хэш-функции алгоритма MD5.

# Class diagram

#### DiscObject

- root: string
- name: string
- create\_time: string
- + do\_type: string

DiscObject()

DiscObject(string root, string name

- + getRoot(): string
- + getName(): string
- + getCreateTime(): string

#### MainWindow

- \*ui: Ui::MainWindow
- \*pathDisc: PathDiscObject
- + MainWindow(QWidget \*parent)
- init\_treewidget()
- show\_path(QTreeWidgetItem \*pTWItem, DiscObject \*pathDisc)
- find\_duplicate()

#### PathDiscObject

- discobjects: list<DiscObject>
- + getDiscObjects() const: list<DiscObject>

# calcDiscObject(): void

#### FileDiscObject

- size: long
- + getSize(): long

### Software tricks

```
void MainWindow::show_path(QTreeWidgetItem*pTWItem3, DiscObject
*pathDisc){
  PathDiscObject dd(pathDisc->getRoot(), pathDisc->getName());
  std::list<DiscObject> discs = dd.getDiscObjects();
  for(std::/list<DiscObject>::iterator obj = discs.begin(); obj != discs.end();
obj++)
    QTreeWidgetItem* pTWItem4 = new QTreeWidgetItem(pTWItem3);
    pTWItem4->setText(0, QString((*obj).getName().data()));
    pTWItem4->setText(1, (*obj).do_type.data());
    if((*obj).do_type.compare("Dir")==0)
      show_path(pTWItem4, &(*obj));
```

