PHOTON

MATF 2020

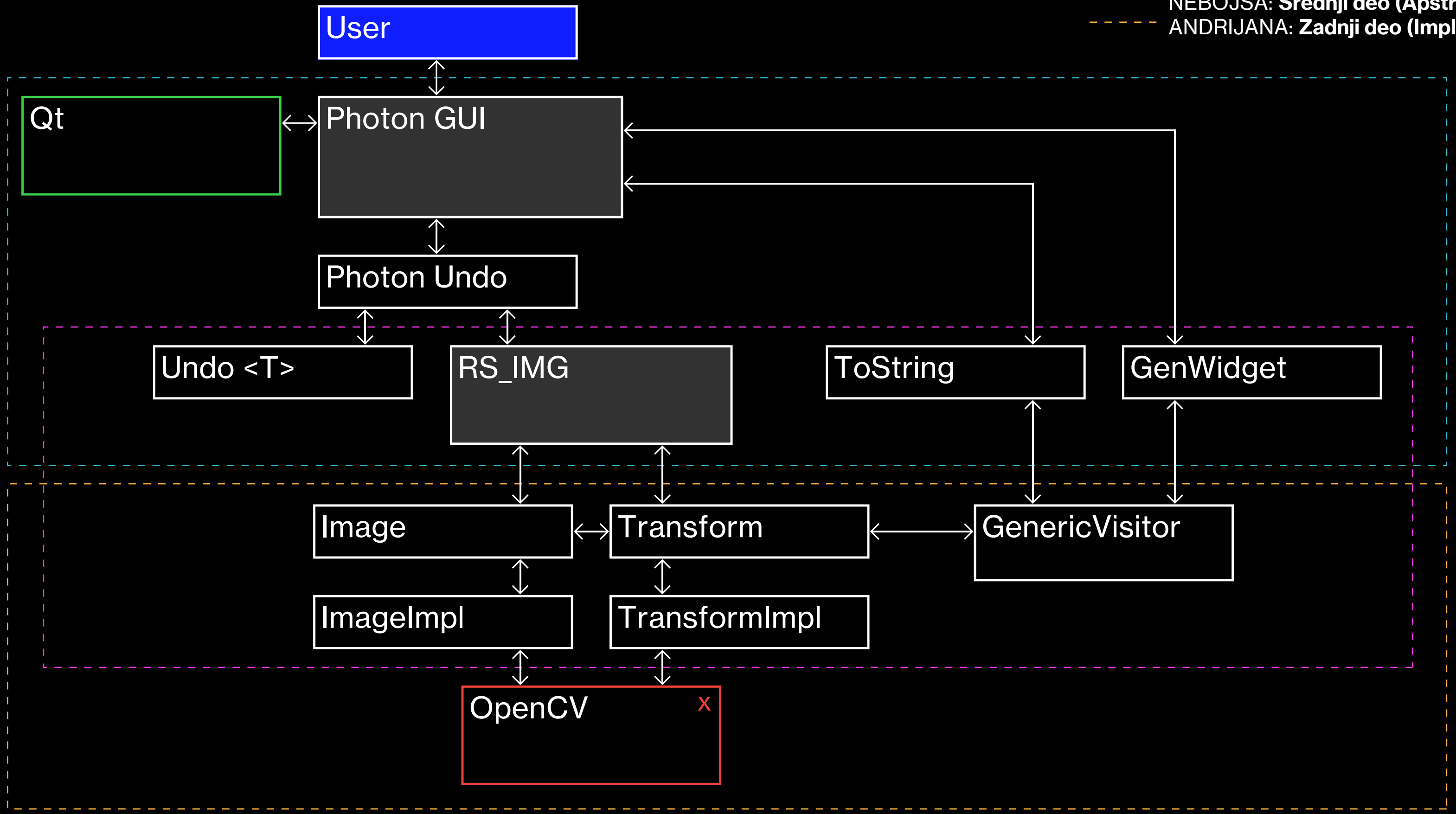
Razvoj softvera

STUDENTI:
Nebojša Koturović
Andrijana Čakarević
Miloš Krsmanović

PROFESOR:
Saša Malkov

ASISTENTI:
Nemanja Mićović
Nikola Ajzenhamer

MILOŠ: **Prednji deo (GUI)**
NEBOJŠA: **Srednji deo (Apstrakcije)**
ANDRIJANA: **Zadnji deo (Implementacije)**



Zadnji deo (Implementacija)

ANDRIJANA

```
// Privatna implementacija: Image.cpp
void Image::write(std::string image_path) const {
    this->impl().write(std::move(image_path));
}

/* Transform.cpp: Implementacija transformacija u OpenCV (Back End-u) */
Image& Rotate::applyImpl(Image &img) const {
    cv::Mat &imageRef = get_impl(img).m_image;
    switch (m_direction) {
        case Direction::RIGHT : cv::rotate( ... ); break;
        case Direction::LEFT : cv::rotate( ... ); break;
    }
    return img;
}
```

Srednji deo (Apstrakcija)

NEBOJŠA

```
Image img("lena.png");

Image rotatedCopy = img << Rotate(Rotate::Direction::LEFT);

/* Rotate u mestu */
img <<= Rotate(Rotate::Direction::RIGHT);

/* Kompozicija transformacija */
img <<= Rotate(Rotate::Direction::LEFT)
     * BlackNWhite()
     * Flip(Flip::Axis::X);

/* Ekvivalentno kompoziciji iznad */
img <<= ( Flip(Flip::Axis::X))
        | BlackNWhite()
        | Rotate(Rotate::Direction::LEFT );
```

Prednji deo (GUI)

MILOŠ

```
void photon_main::on_actionRotacija_ulevo_triggered() {
    m_imageUndo.action(rs::Rotate(rs::Rotate::Direction::LEFT));
    ui->slika->update();
}

void photon_main::on_undo_triggered() {
    m_imageUndo.undo(); ui->slika->update();
}

void photon_main::on_undo_triggered() {
    WidgetGenerator widgetGen; // Visitor

    for (auto e : m_imageUndo.history())
        histWidget.add(widgetGen(e));

    ui->istorija->update();
}
```

Undo <T>

```
/**** Undo istorija za ceo broj ****/
Undo<int> intUndo(2);

intUndo.action([](int &refToVal) { refToVal += 2; }
std::cout << intUndo.current(); // prints 4
std::cout << intUndo.origin(); // prints 2

intUndo.undo(); // current = 2
intUndo.redo(); // current = 4
intUndo.history(); // { 2, 4 };

/**** Undo istorija za Image ****/
Undo<Image, Transform> imgUndo(std::move(img));
imgUndo.action(Rotate(Rotate::Direction::RIGHT);
imgUndo.undo(); // Before rotation
imgUndo.redo(); // With rotation
imgUndo.history(); // { ORIGINAL, ROTATED }
```

Generički Visitor Pattern

```cpp
/* Genericki Visitor Pattern */
#include "visitor_pattern.hpp"

#define TypeHierarchy Transform, BlackNWhite, Flip
using MyConstVisitorI = Visitor<ConstTypeHierarchy>;

/************** Bazna klasa **************/
class Transform : public Visitable<TypeHierarchy> {};
/***************** Izvedene klase ********************/
class Rotate : public InheritVisitable<Rotate,Transform,TypeHierarchy> { ... };
class BlackNWhite : public InheritVisitable<BlackNWhite,Transform,TypeHierarchy> { ... };
class Flip : public InheritVisitable<Flip,Transform,TypeHierarchy> { ... };

class CustomVisitor : MyConstVisitorI {
   void visit(Rotate &r) { ... };
   void visit(BlackNWhite &bw) { ... };
   void otherwise(Transform &t) { ... };
} customVisitor;
customVisitor(refToBase); // -> Poziva odgovarajuci overload ili otherwise
```

Hvala na pažnji.