Project1 보고서

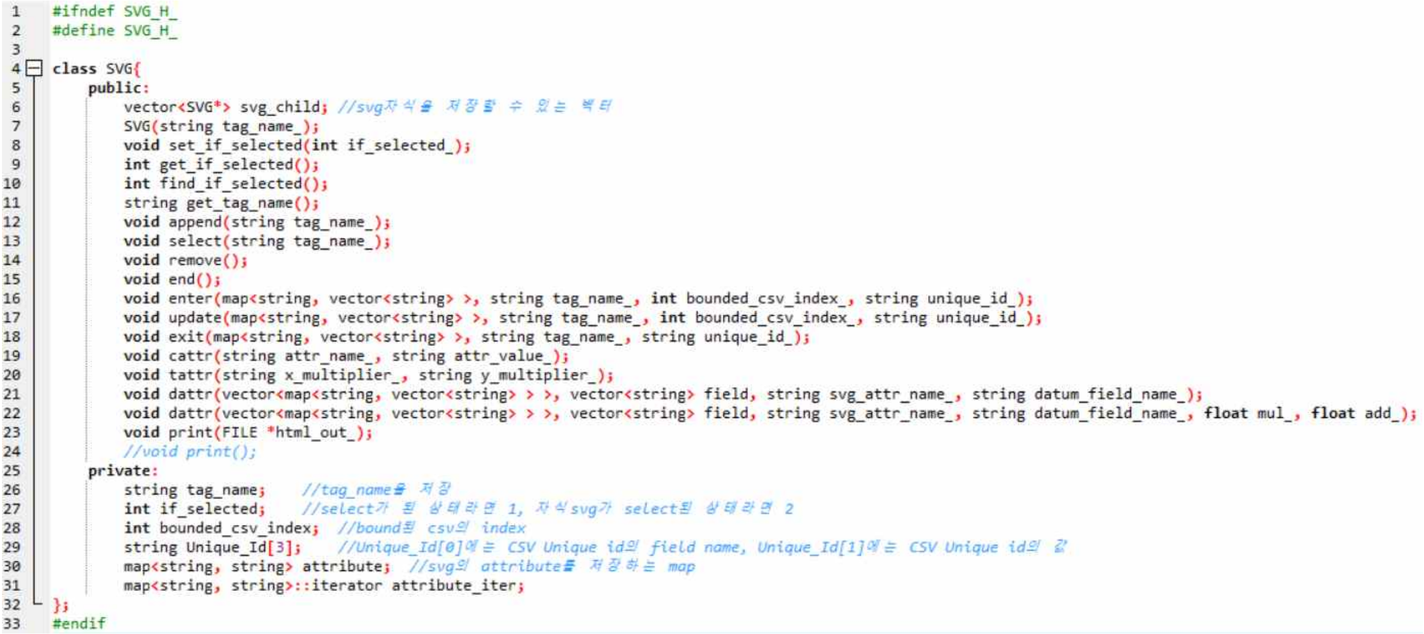
컴퓨터공학부
2016-12299 박용주

Environment

window 10, Dev-C++

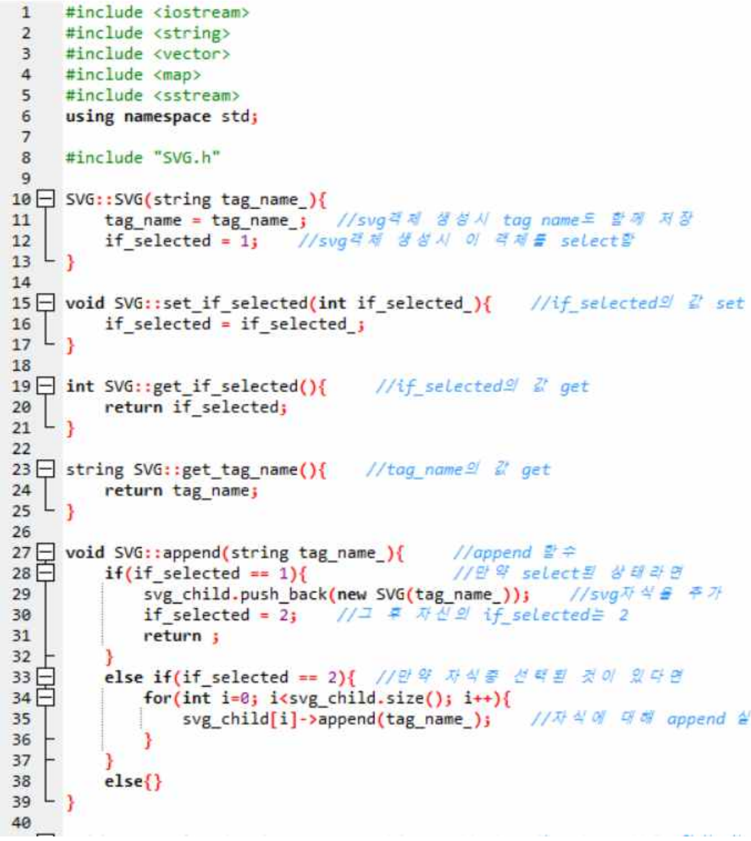
1. code and explanation(주석처리)

1)svg.h

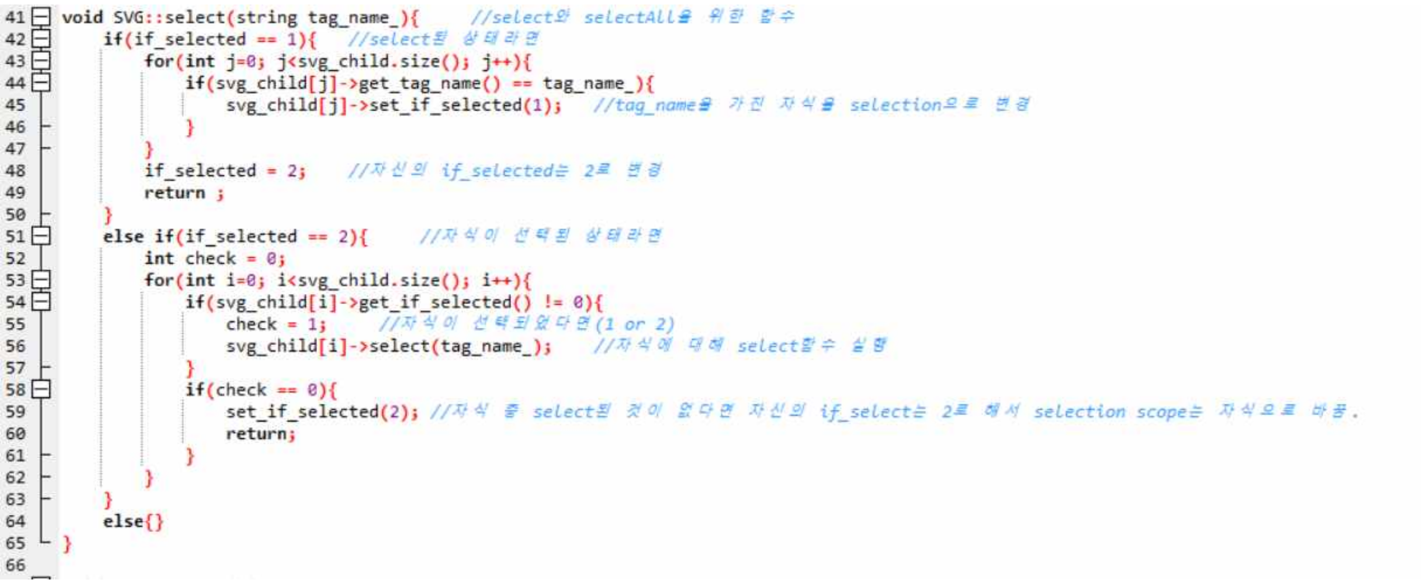


```cpp
#ifndef SVG_H_
#define SVG_H_

class SVG{
    public:
        vector<SVG*> svg_child; //svg자식을 저장할 수 있는 벡터
        SVG(string tag_name_);
        void set_if_selected(int if_selected_);
        int get_if_selected();
        int find_if_selected();
        string get_tag_name();
        void append(string tag_name_);
        void select(string tag_name_);
        void remove();
        void end();
        void enter(map<string, vector<string> >, string tag_name_, int bounded_csv_index_, string unique_id_);
        void update(map<string, vector<string> >, string tag_name_, int bounded_csv_index_, string unique_id_);
        void exit(map<string, vector<string> >, string tag_name_, string unique_id_);
        void cattr(string attr_name_, string attr_value_);
        void tattr(string x_multiplier_, string y_multiplier_);
        void dattr(vector<map<string, vector<string> > >, vector<string> field, string svg_attr_name_, string datum_field_name_);
        void dattr(vector<map<string, vector<string> > >, vector<string> field, string svg_attr_name_, string datum_field_name_, float mul_, float add_);
        void print(FILE *html_out_);
        //void print();
    private:
        string tag_name;    //tag_name을 저장
        int if_selected;    //select가 된 상태라면 1, 자식svg가 select된 상태라면 2
        int bounded_csv_index;  //bound된 csv의 index
        string Unique_Id[3];    //Unique_Id[0]에는 CSV Unique id의 field name, Unique_Id[1]에는 CSV Unique id의 값
        map<string, string> attribute;  //svg의 attribute를 저장하는 map
        map<string, string>::iterator attribute_iter;
};
#endif
```

2)svg.cpp

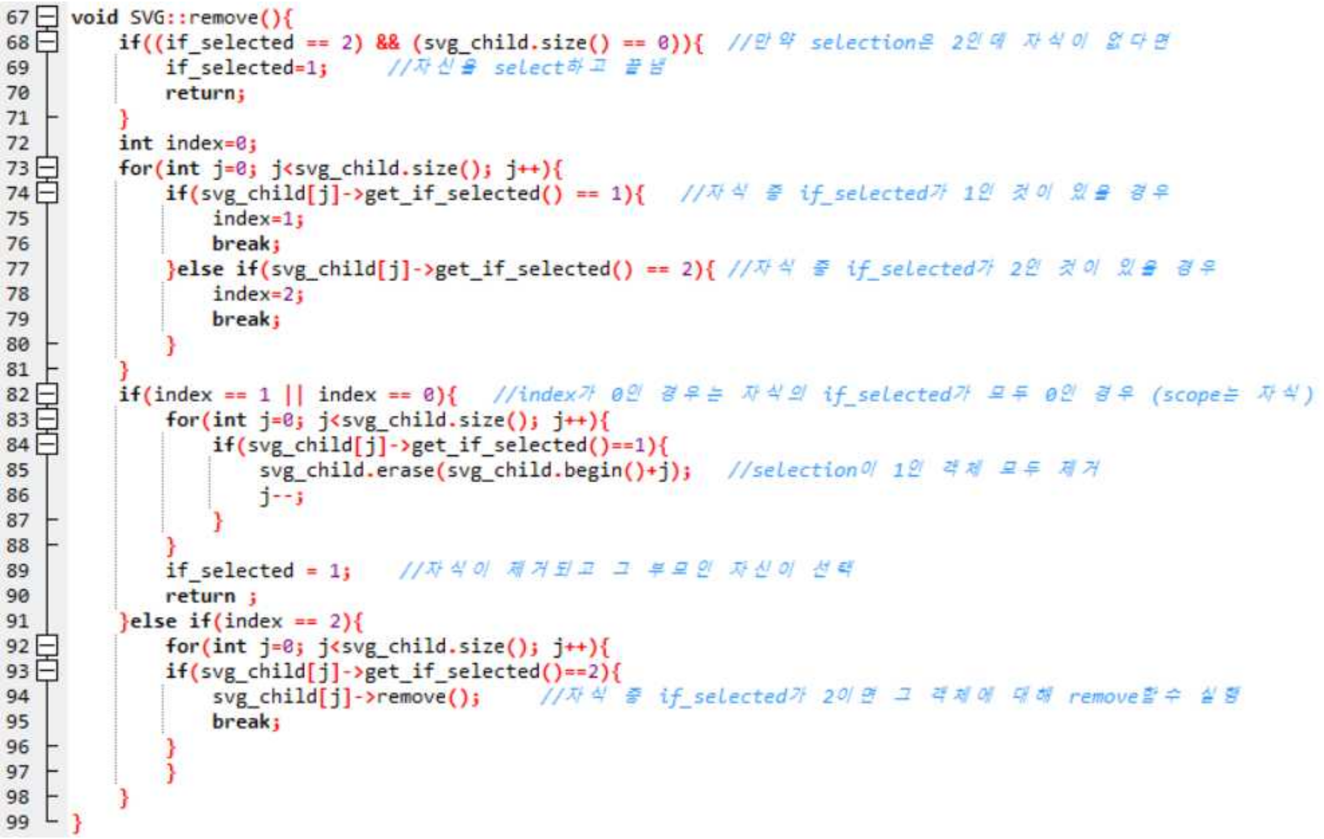


```cpp
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include <sstream>
using namespace std;

#include "SVG.h"

SVG::SVG(string tag_name_){
    tag_name = tag_name_;   //svg객체 생성시 tag name도 함께 저장
    if_selected = 1;    //svg객체 생성시 이 객체를 select함
}

void SVG::set_if_selected(int if_selected_){    //if_selected의 값 set
    if_selected = if_selected_;
}

int SVG::get_if_selected(){     //if_selected의 값 get
    return if_selected;
}

string SVG::get_tag_name(){    //tag_name의 값 get
    return tag_name;
}

void SVG::append(string tag_name_){     //append 함수
    if(if_selected == 1){               //만약 select된 상태라면
        svg_child.push_back(new SVG(tag_name_));    //svg자식을 추가
        if_selected = 2;    //그 후 자신의 if_selected는 2
        return ;
    }
    else if(if_selected == 2){ //만약 자식중 선택된 것이 있다면
        for(int i=0; i<svg_child.size(); i++){
            svg_child[i]->append(tag_name_);    //자식에 대해 append 실행
        }
    }
    else{}
}

```



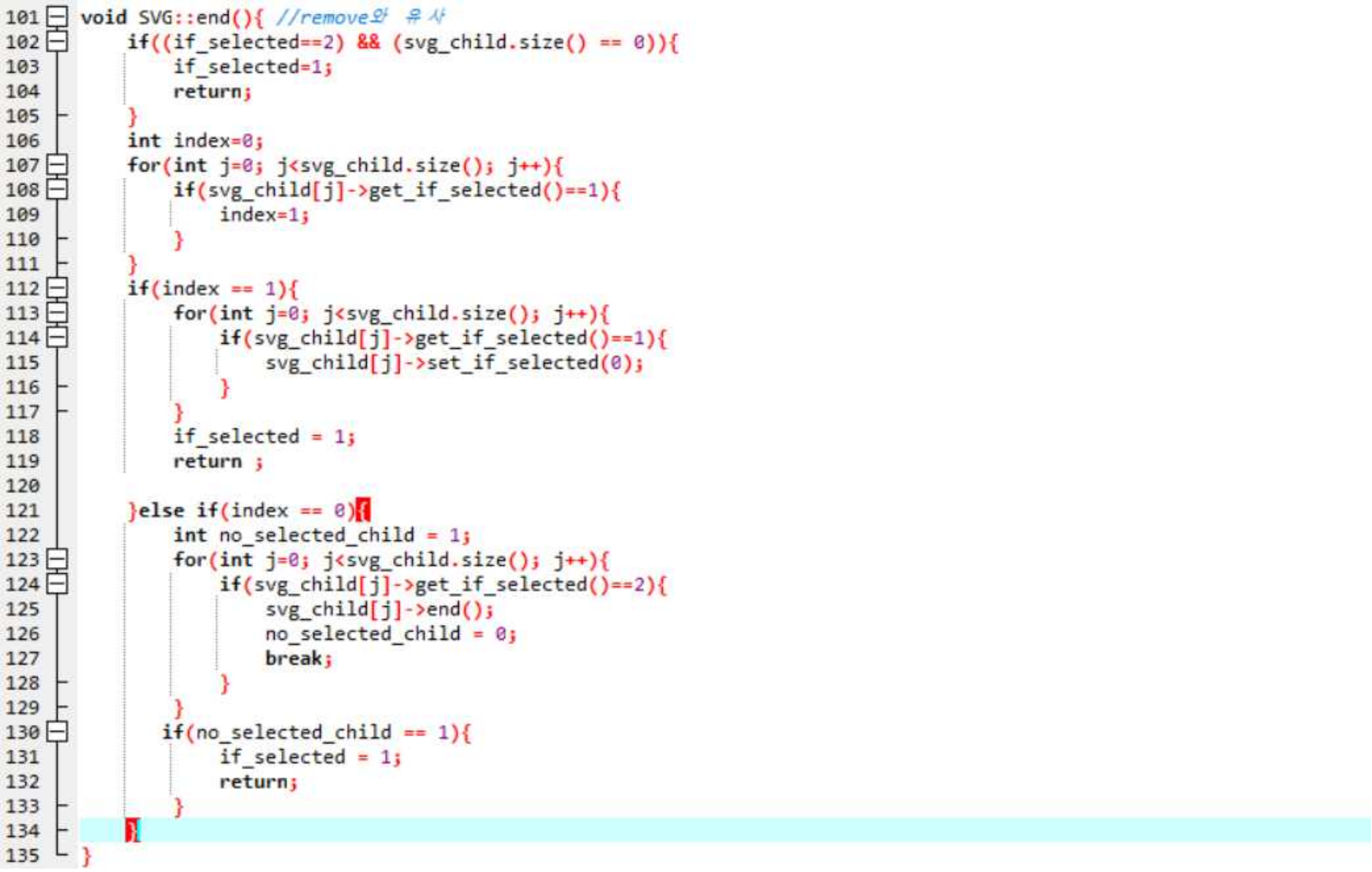
```cpp
void SVG::select(string tag_name_){     //select와 selectAll을 위한 함수
    if(if_selected == 1){   //select된 상태라면
        for(int j=0; j<svg_child.size(); j++){
            if(svg_child[j]->get_tag_name() == tag_name_){
                svg_child[j]->set_if_selected(1);   //tag_name을 가진 자식을 selection으로 변경
            }
        }
        if_selected = 2;    //자신의 if_selected는 2로 변경
        return ;
    }
    else if(if_selected == 2){     //자식이 선택된 상태라면
        int check = 0;
        for(int i=0; i<svg_child.size(); i++){
            if(svg_child[i]->get_if_selected() != 0){
                check = 1;     //자식이 선택되었다면(1 or 2)
                svg_child[i]->select(tag_name_);    //자식에 대해 select함수 실행
            }
            if(check == 0){
                set_if_selected(2); //자식 중 select된 것이 없다면 자신의 if_select는 2로 해서 selection scope는 자식으로 바꿈.
                return;
            }
        }
    }
    else{}
}
```

그림 3 select



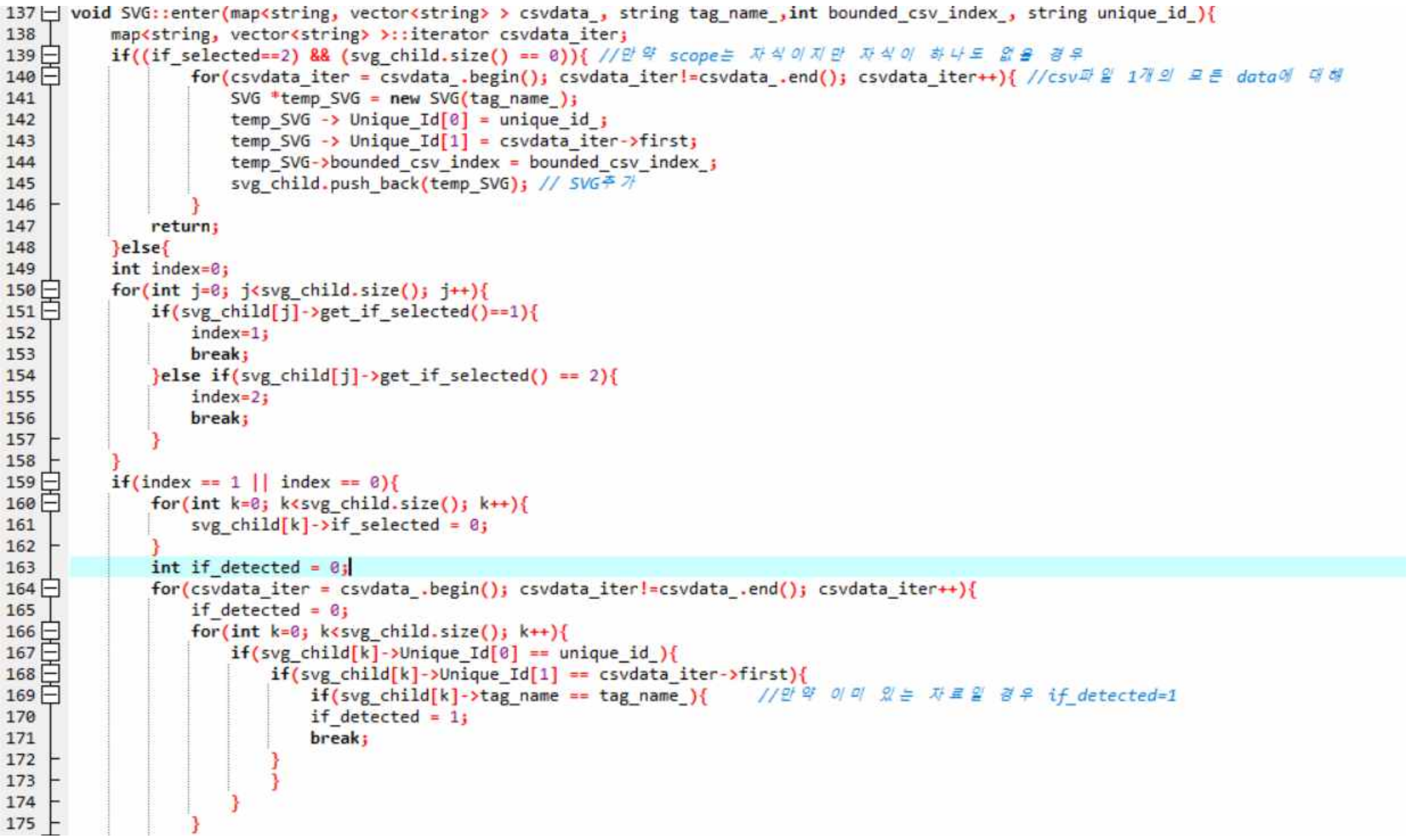
```cpp
void SVG::remove(){
    if((if_selected == 2) && (svg_child.size() == 0)){ //만약 selection은 2인데 자식이 없다면
        if_selected=1;     //자신을 select하고 끝냄
        return;
    }
    int index=0;
    for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected() == 1){   //자식 중 if_selected가 1인 것이 있을 경우
            index=1;
            break;
        }else if(svg_child[j]->get_if_selected() == 2){ //자식 중 if_selected가 2인 것이 있을 경우
            index=2;
            break;
        }
    }
    if(index == 1 || index == 0){   //index가 0인 경우는 자식의 if_selected가 모두 0인 경우 (scope는 자식)
        for(int j=0; j<svg_child.size(); j++){
            if(svg_child[j]->get_if_selected()==1){
                svg_child.erase(svg_child.begin()+j);   //selection이 1인 객체 모두 제거
                j--;
            }
        }
        if_selected = 1;    //자식이 제거되고 그 부모인 자신이 선택
        return ;
    }else if(index == 2){
        for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected()==2){
            svg_child[j]->remove();     //자식 중 if_selected가 2이면 그 객체에 대해 remove함수 실행
            break;
        }
        }
    }
}
```

그림 4 remove

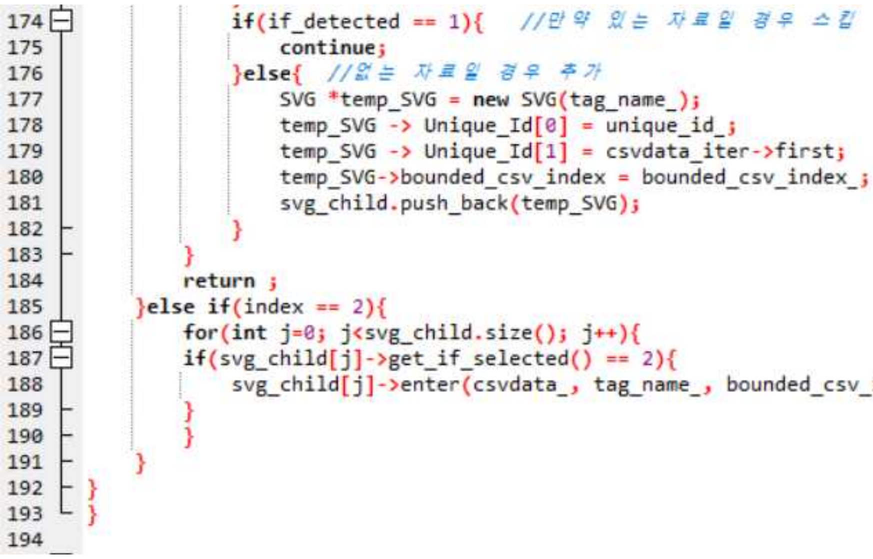


```
void SVG::end(){ //remove와 유사
    if((if_selected==2) && (svg_child.size() == 0)){
        if_selected=1;
        return;
    }
    int index=0;
    for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected()==1){
            index=1;
        }
    }
    if(index == 1){
        for(int j=0; j<svg_child.size(); j++){
            if(svg_child[j]->get_if_selected()==1){
                svg_child[j]->set_if_selected(0);
            }
        }
        if_selected = 1;
        return ;

    }else if(index == 0){
        int no_selected_child = 1;
        for(int j=0; j<svg_child.size(); j++){
            if(svg_child[j]->get_if_selected()==2){
                svg_child[j]->end();
                no_selected_child = 0;
                break;
            }
        }
       if(no_selected_child == 1){
            if_selected = 1;
            return;
        }
    }
}
```

그림 5 end

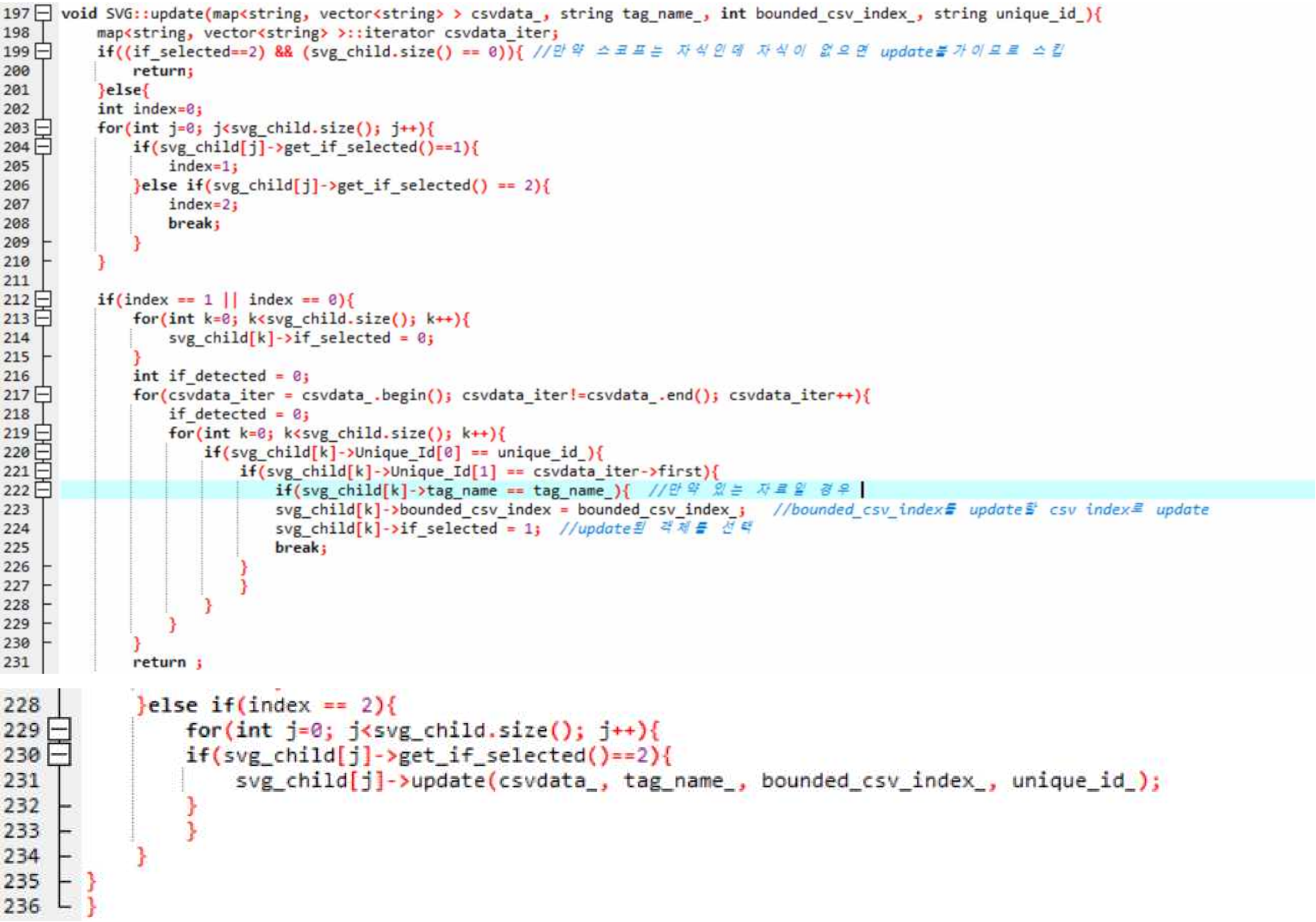


```
void SVG::enter(map<string, vector<string> > csvdata_, string tag_name_,int bounded_csv_index_, string unique_id_){
    map<string, vector<string> >::iterator csvdata_iter;
    if((if_selected==2) && (svg_child.size() == 0)){ //만약 scope는 자식이지만 자식이 하나도 없을 경우
            for(csvdata_iter = csvdata_.begin(); csvdata_iter!=csvdata_.end(); csvdata_iter++){ //csv파일 1개의 모든 data에 대해
                SVG *temp_SVG = new SVG(tag_name_);
                temp_SVG -> Unique_Id[0] = unique_id_;
                temp_SVG -> Unique_Id[1] = csvdata_iter->first;
                temp_SVG->bounded_csv_index = bounded_csv_index_;
                svg_child.push_back(temp_SVG); // SVG추가
            }
        return;
    }else{
    int index=0;
    for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected()==1){
            index=1;
            break;
        }else if(svg_child[j]->get_if_selected() == 2){
            index=2;
            break;
        }
    }
    if(index == 1 || index == 0){
        for(int k=0; k<svg_child.size(); k++){
            svg_child[k]->if_selected = 0;
        }
        int if_detected = 0;
        for(csvdata_iter = csvdata_.begin(); csvdata_iter!=csvdata_.end(); csvdata_iter++){
            if_detected = 0;
            for(int k=0; k<svg_child.size(); k++){
                if(svg_child[k]->Unique_Id[0] == unique_id_){
                    if(svg_child[k]->Unique_Id[1] == csvdata_iter->first){
                        if(svg_child[k]->tag_name == tag_name_){     //만약 이미 있는 자료일 경우 if_detected=1
                        if_detected = 1;
                        break;
                    }
                    }
                }
            }
```



```
            if(if_detected == 1){   //만약 있는 자료일 경우 스킵
                continue;
            }else{  //없는 자료일 경우 추가
                SVG *temp_SVG = new SVG(tag_name_);
                temp_SVG -> Unique_Id[0] = unique_id_;
                temp_SVG -> Unique_Id[1] = csvdata_iter->first;
                temp_SVG->bounded_csv_index = bounded_csv_index_;
                svg_child.push_back(temp_SVG);
            }
        }
        return ;
    }else if(index == 2){
        for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected() == 2){
            svg_child[j]->enter(csvdata_, tag_name_, bounded_csv_index_, unique_id_);
        }
        }
    }
}
}
```

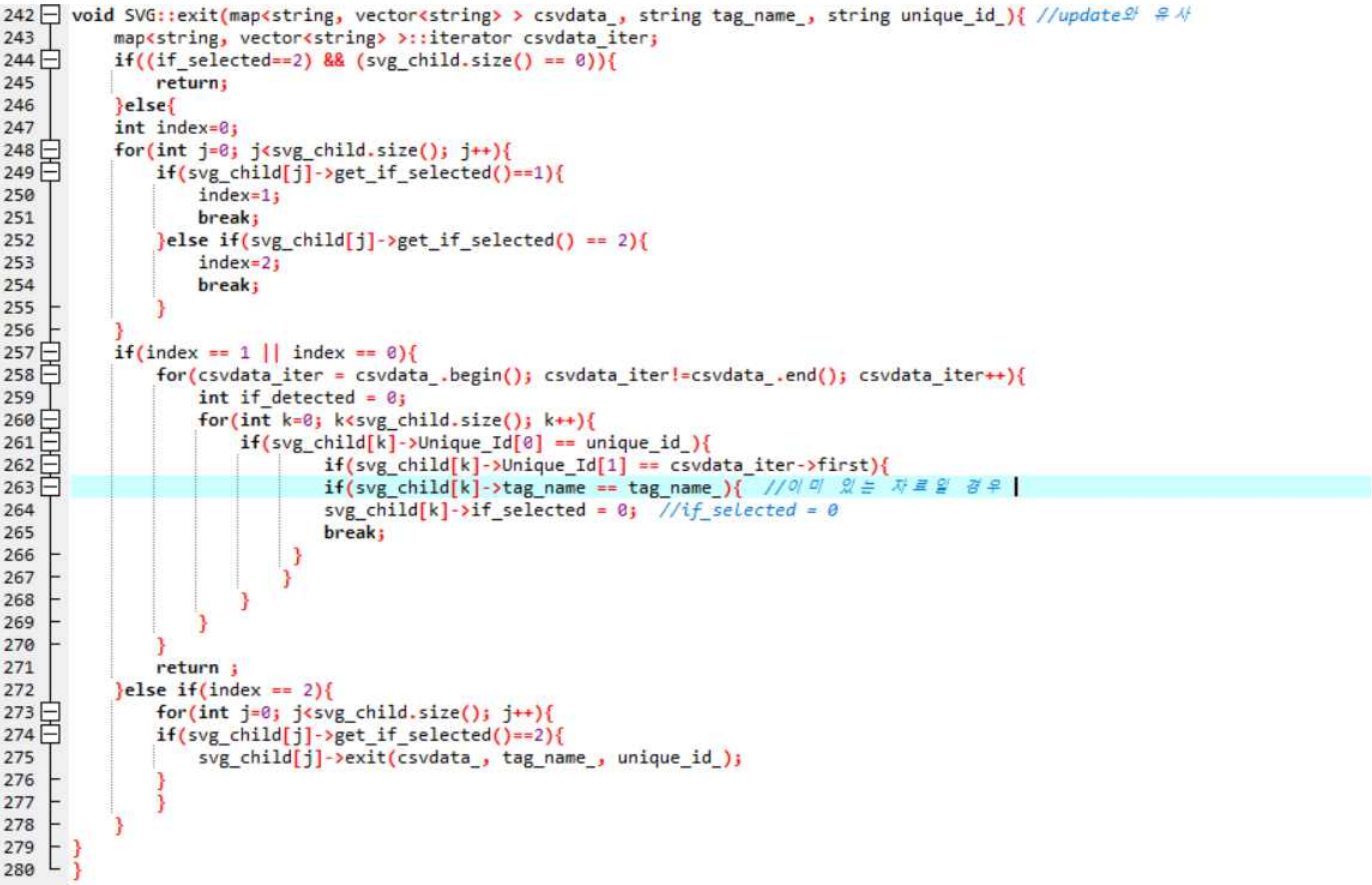
그림 7 enter



```
void SVG::update(map<string, vector<string> > csvdata_, string tag_name_, int bounded_csv_index_, string unique_id_){
    map<string, vector<string> >::iterator csvdata_iter;
    if((if_selected==2) && (svg_child.size() == 0)){ //만약 스코프는 자식인데 자식이 없으면 update를가이므로 스킵
        return;
    }else{
    int index=0;
    for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected()==1){
            index=1;
        }else if(svg_child[j]->get_if_selected() == 2){
            index=2;
            break;
        }
    }

    if(index == 1 || index == 0){
        for(int k=0; k<svg_child.size(); k++){
            svg_child[k]->if_selected = 0;
        }
        int if_detected = 0;
        for(csvdata_iter = csvdata_.begin(); csvdata_iter!=csvdata_.end(); csvdata_iter++){
            if_detected = 0;
            for(int k=0; k<svg_child.size(); k++){
                if(svg_child[k]->Unique_Id[0] == unique_id_){
                    if(svg_child[k]->Unique_Id[1] == csvdata_iter->first){
                        if(svg_child[k]->tag_name == tag_name_){  //만약 있는 자료일 경우
                        svg_child[k]->bounded_csv_index = bounded_csv_index_;   //bounded_csv_index를 update될 csv index로 update
                        svg_child[k]->if_selected = 1;  //update된 객체를 선택
                        break;
                    }
                    }
                }
            }
        }
        return ;
```

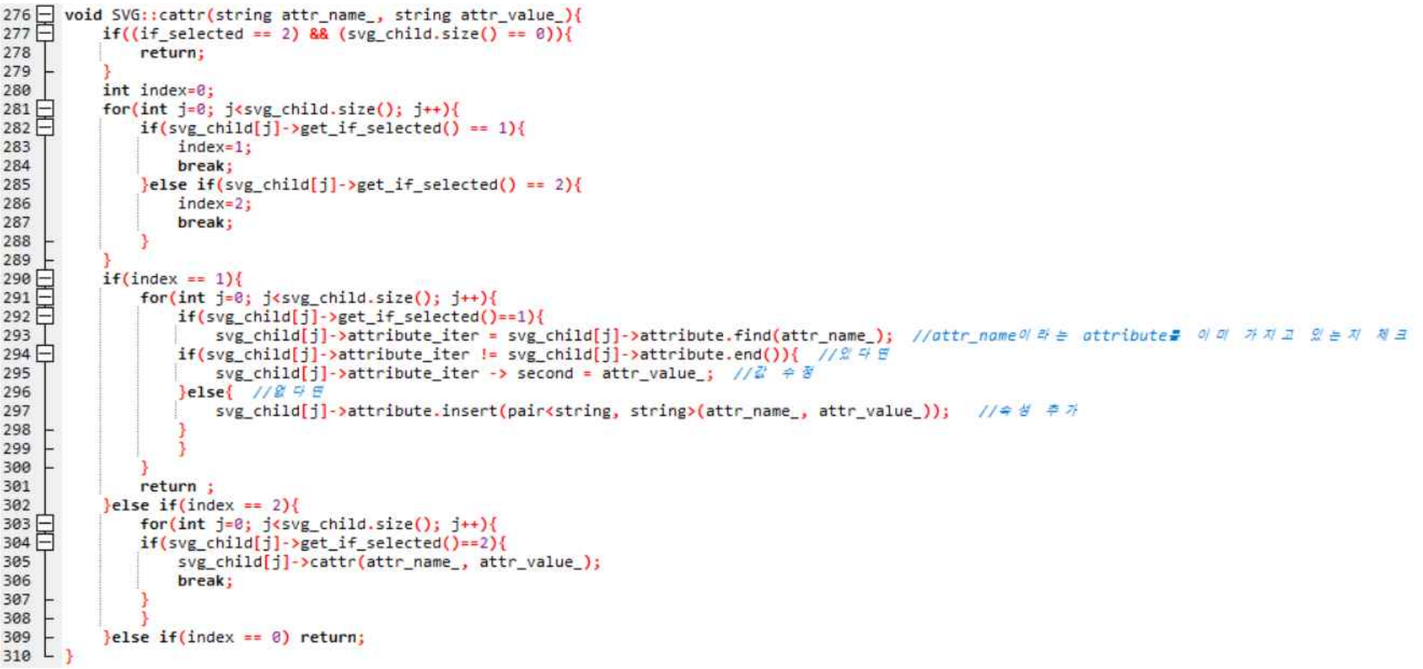
```
    }else if(index == 2){
        for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected()==2){
            svg_child[j]->update(csvdata_, tag_name_, bounded_csv_index_, unique_id_);
        }
        }
    }
}
}
```

그림 9 update



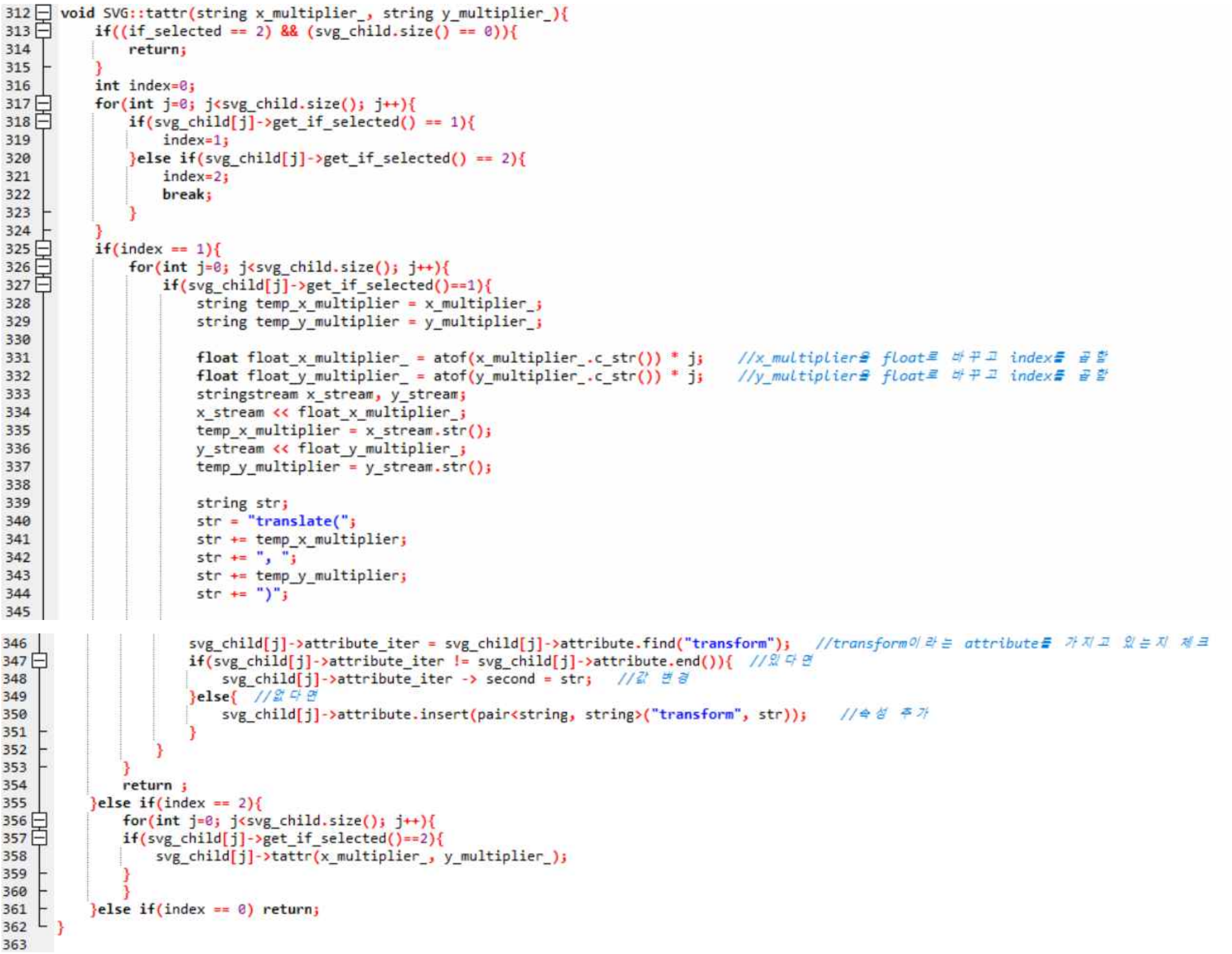
```
void SVG::exit(map<string, vector<string> > csvdata_, string tag_name_, string unique_id_){ //update와 유사
    map<string, vector<string> >::iterator csvdata_iter;
    if((if_selected==2) && (svg_child.size() == 0)){
        return;
    }else{
    int index=0;
    for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected()==1){
            index=1;
            break;
        }else if(svg_child[j]->get_if_selected() == 2){
            index=2;
            break;
        }
    }
    if(index == 1 || index == 0){
        for(csvdata_iter = csvdata_.begin(); csvdata_iter!=csvdata_.end(); csvdata_iter++){
            int if_detected = 0;
            for(int k=0; k<svg_child.size(); k++){
                if(svg_child[k]->Unique_Id[0] == unique_id_){
                        if(svg_child[k]->Unique_Id[1] == csvdata_iter->first){
                        if(svg_child[k]->tag_name == tag_name_){ //이미 있는 자료일 경우
                        svg_child[k]->if_selected = 0;  //if_selected = 0
                        break;
                      }
                     }
                }
            }
        }
        return ;
    }else if(index == 2){
        for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected()==2){
            svg_child[j]->exit(csvdata_, tag_name_, unique_id_);
        }
        }
    }
}
}
```

그림 10 exit



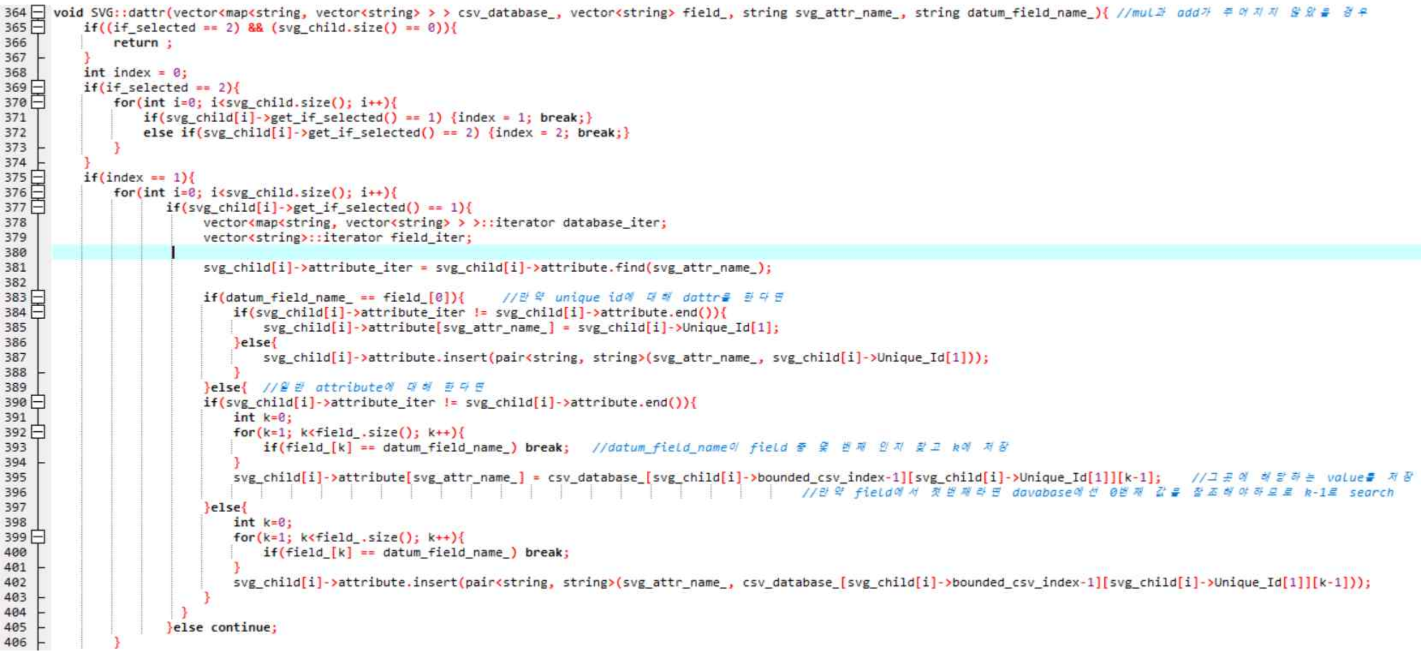
```
void SVG::cattr(string attr_name_, string attr_value_){
    if((if_selected == 2) && (svg_child.size() == 0)){
        return;
    }
    int index=0;
    for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected() == 1){
            index=1;
            break;
        }else if(svg_child[j]->get_if_selected() == 2){
            index=2;
            break;
        }
    }
    if(index == 1){
        for(int j=0; j<svg_child.size(); j++){
            if(svg_child[j]->get_if_selected()==1){
                svg_child[j]->attribute_iter = svg_child[j]->attribute.find(attr_name_);  //attr_name이라는 attribute를 이미 가지고 있는지 체크
            if(svg_child[j]->attribute_iter != svg_child[j]->attribute.end()){  //있다면
                svg_child[j]->attribute_iter -> second = attr_value_;  //값 수정
            }else{  //없다면
                svg_child[j]->attribute.insert(pair<string, string>(attr_name_, attr_value_));  //속성 추가
            }
            }
        }
        return ;
    }else if(index == 2){
        for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected()==2){
            svg_child[j]->cattr(attr_name_, attr_value_);
            break;
        }
        }
    }else if(index == 0) return;
}
```

그림 11 cattr

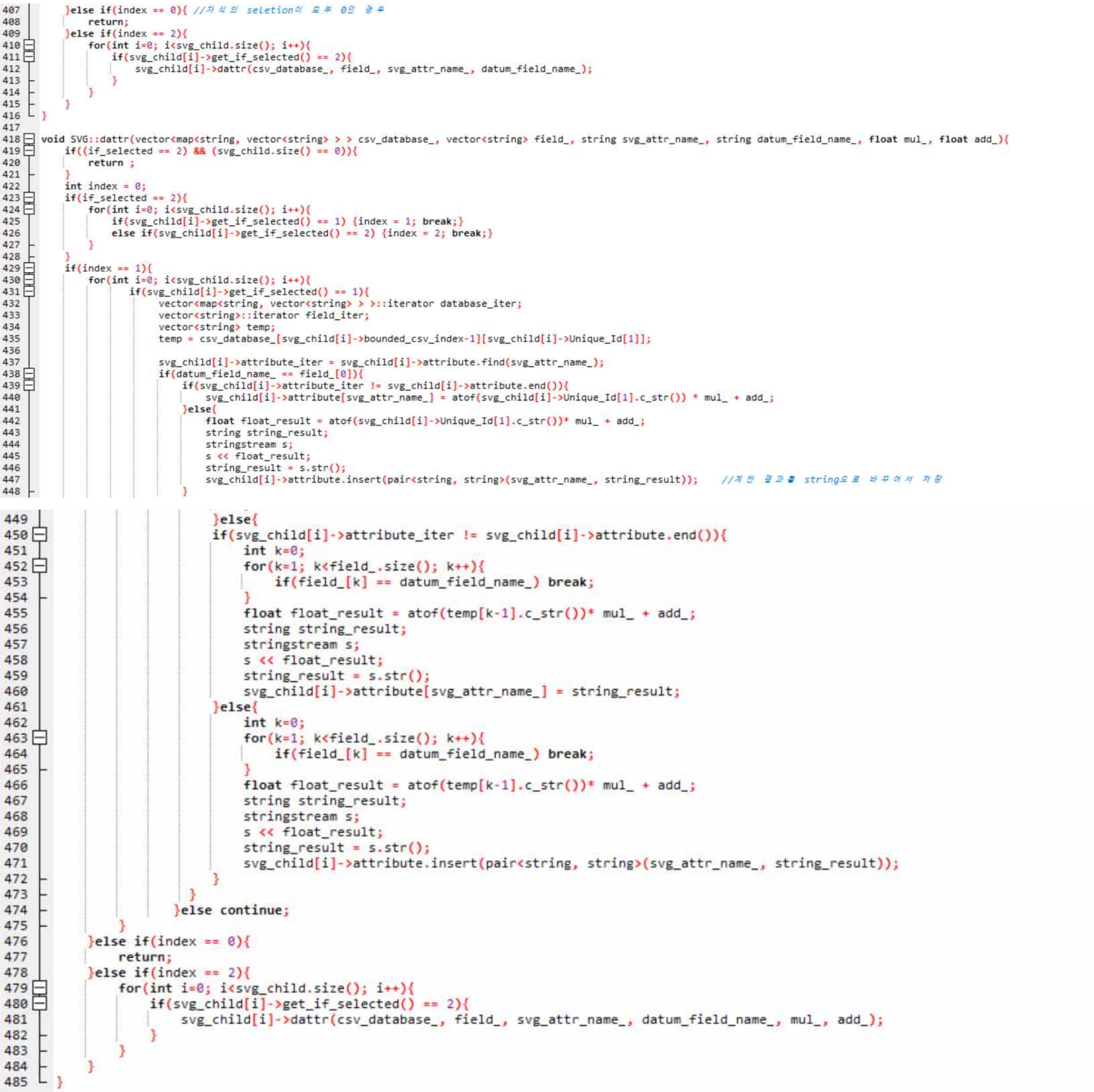


```
void SVG::tattr(string x_multiplier_, string y_multiplier_){
    if((if_selected == 2) && (svg_child.size() == 0)){
        return;
    }
    int index=0;
    for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected() == 1){
            index=1;
        }else if(svg_child[j]->get_if_selected() == 2){
            index=2;
            break;
        }
    }
    if(index == 1){
        for(int j=0; j<svg_child.size(); j++){
            if(svg_child[j]->get_if_selected()==1){
                string temp_x_multiplier = x_multiplier_;
                string temp_y_multiplier = y_multiplier_;

                float float_x_multiplier_ = atof(x_multiplier_.c_str()) * j;    //x_multiplier를 float로 바꾸고 index를 곱함
                float float_y_multiplier_ = atof(y_multiplier_.c_str()) * j;    //y_multiplier를 float로 바꾸고 index를 곱함
                stringstream x_stream, y_stream;
                x_stream << float_x_multiplier_;
                temp_x_multiplier = x_stream.str();
                y_stream << float_y_multiplier_;
                temp_y_multiplier = y_stream.str();

                string str;
                str = "translate(";
                str += temp_x_multiplier;
                str += ", ";
                str += temp_y_multiplier;
                str += ")";

               svg_child[j]->attribute_iter = svg_child[j]->attribute.find("transform");   //transform이라는 attribute를 가지고 있는지 체크
               if(svg_child[j]->attribute_iter != svg_child[j]->attribute.end()){  //있다면
                   svg_child[j]->attribute_iter -> second = str;   //값 변경
               }else{  //없다면
                   svg_child[j]->attribute.insert(pair<string, string>("transform", str));    //속성 추가
               }
            }
        }
        return ;
    }else if(index == 2){
        for(int j=0; j<svg_child.size(); j++){
        if(svg_child[j]->get_if_selected()==2){
            svg_child[j]->tattr(x_multiplier_, y_multiplier_);
        }
        }
    }else if(index == 0) return;
}
```

그림 13 tattr

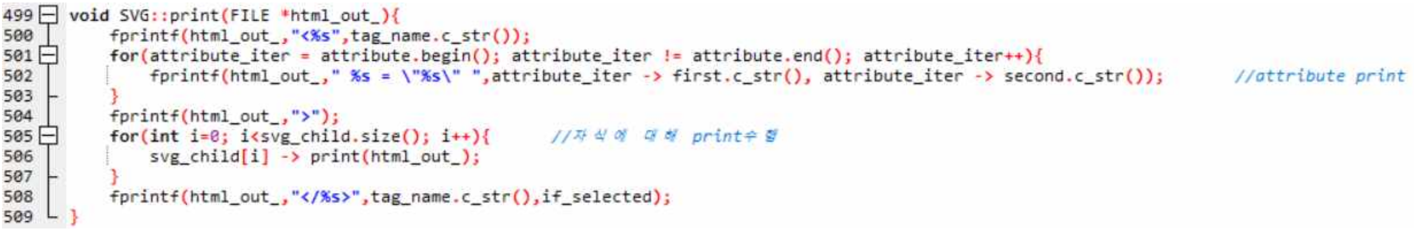


```
void SVG::dattr(vector<map<string, vector<string> > > csv_database_, vector<string> field_, string svg_attr_name_, string datum_field_name_){ //mul과 add가 주어지지 않았을 경우
    if((if_selected == 2) && (svg_child.size() == 0)){
        return ;
    }
    int index = 0;
    if(if_selected == 2){
        for(int i=0; i<svg_child.size(); i++){
            if(svg_child[i]->get_if_selected() == 1) {index = 1; break;}
            else if(svg_child[i]->get_if_selected() == 2) {index = 2; break;}
        }
    }
    if(index == 1){
        for(int i=0; i<svg_child.size(); i++){
                if(svg_child[i]->get_if_selected() == 1){
                    vector<map<string, vector<string> > >::iterator database_iter;
                    vector<string>::iterator field_iter;

                    svg_child[i]->attribute_iter = svg_child[i]->attribute.find(svg_attr_name_);

                    if(datum_field_name_ == field_[0]){     //만약 unique id에 대해 dattr를 한다면
                        if(svg_child[i]->attribute_iter != svg_child[i]->attribute.end()){
                            svg_child[i]->attribute[svg_attr_name_] = svg_child[i]->Unique_Id[1];
                        }else{
                            svg_child[i]->attribute.insert(pair<string, string>(svg_attr_name_, svg_child[i]->Unique_Id[1]));
                        }
                    }else{  //일반 attribute에 대해 한다면
                    if(svg_child[i]->attribute_iter != svg_child[i]->attribute.end()){
                        int k=0;
                        for(k=1; k<field_.size(); k++){
                            if(field_[k] == datum_field_name_) break;   //datum_field_name이 field 중 몇 번째 인지 찾고 k에 저장
                        }
                        svg_child[i]->attribute[svg_attr_name_] = csv_database_[svg_child[i]->bounded_csv_index-1][svg_child[i]->Unique_Id[1]][k-1];    //그곳에 해당하는 value를 저장
                                                                                              //만약 field에서 첫번째라면 davabase에선 0번째 값을 참조해야하므로 k-1로 search
                    }else{
                        int k=0;
                        for(k=1; k<field_.size(); k++){
                            if(field_[k] == datum_field_name_) break;
                        }
                        svg_child[i]->attribute.insert(pair<string, string>(svg_attr_name_, csv_database_[svg_child[i]->bounded_csv_index-1][svg_child[i]->Unique_Id[1]][k-1]));
                    }
                }
            }else continue;
        }
```



```
    }else if(index == 0){ //자식의 seletion이 모두 0인 경우
        return;
    }else if(index == 2){
        for(int i=0; i<svg_child.size(); i++){
            if(svg_child[i]->get_if_selected() == 2){
                svg_child[i]->dattr(csv_database_, field_, svg_attr_name_, datum_field_name_);
            }
        }
    }
}

void SVG::dattr(vector<map<string, vector<string> > > csv_database_, vector<string> field_, string svg_attr_name_, string datum_field_name_, float mul_, float add_){
    if((if_selected == 2) && (svg_child.size() == 0)){
        return ;
    }
    int index = 0;
    if(if_selected == 2){
        for(int i=0; i<svg_child.size(); i++){
            if(svg_child[i]->get_if_selected() == 1) {index = 1; break;}
            else if(svg_child[i]->get_if_selected() == 2) {index = 2; break;}
        }
    }
    if(index == 1){
        for(int i=0; i<svg_child.size(); i++){
            if(svg_child[i]->get_if_selected() == 1){
                vector<map<string, vector<string> > >::iterator database_iter;
                vector<string>::iterator field_iter;
                vector<string> temp;
                temp = csv_database_[svg_child[i]->bounded_csv_index-1][svg_child[i]->Unique_Id[1]];

                svg_child[i]->attribute_iter = svg_child[i]->attribute.find(svg_attr_name_);
                if(datum_field_name_ == field_[0]){
                    if(svg_child[i]->attribute_iter != svg_child[i]->attribute.end()){
                        svg_child[i]->attribute[svg_attr_name_] = atof(svg_child[i]->Unique_Id[1].c_str()) * mul_ + add_;
                    }else{
                        float float_result = atof(svg_child[i]->Unique_Id[1].c_str())* mul_ + add_;
                        string string_result;
                        stringstream s;
                        s << float_result;
                        string_result = s.str();
                        svg_child[i]->attribute.insert(pair<string, string>(svg_attr_name_, string_result));    //계산 결과를 string으로 바꾸어서 저장
                    }
                }else{
                if(svg_child[i]->attribute_iter != svg_child[i]->attribute.end()){
                    int k=0;
                    for(k=1; k<field_.size(); k++){
                        if(field_[k] == datum_field_name_) break;
                    }
                    float float_result = atof(temp[k-1].c_str())* mul_ + add_;
                    string string_result;
                    stringstream s;
                    s << float_result;
                    string_result = s.str();
                    svg_child[i]->attribute[svg_attr_name_] = string_result;
                }else{
                    int k=0;
                    for(k=1; k<field_.size(); k++){
                        if(field_[k] == datum_field_name_) break;
                    }
                    float float_result = atof(temp[k-1].c_str())* mul_ + add_;
                    string string_result;
                    stringstream s;
                    s << float_result;
                    string_result = s.str();
                    svg_child[i]->attribute.insert(pair<string, string>(svg_attr_name_, string_result));
                }
                }
            }else continue;
        }
    }else if(index == 0){
        return;
    }else if(index == 2){
        for(int i=0; i<svg_child.size(); i++){
            if(svg_child[i]->get_if_selected() == 2){
                svg_child[i]->dattr(csv_database_, field_, svg_attr_name_, datum_field_name_, mul_, add_);
            }
        }
    }
}
```

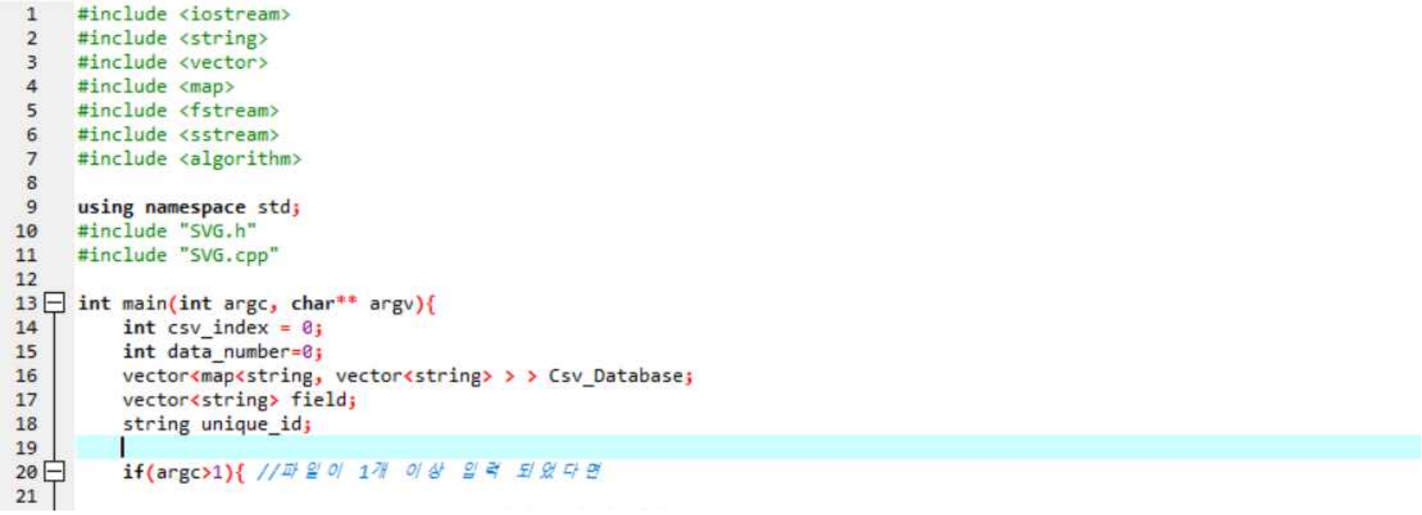
그림 16 dattr



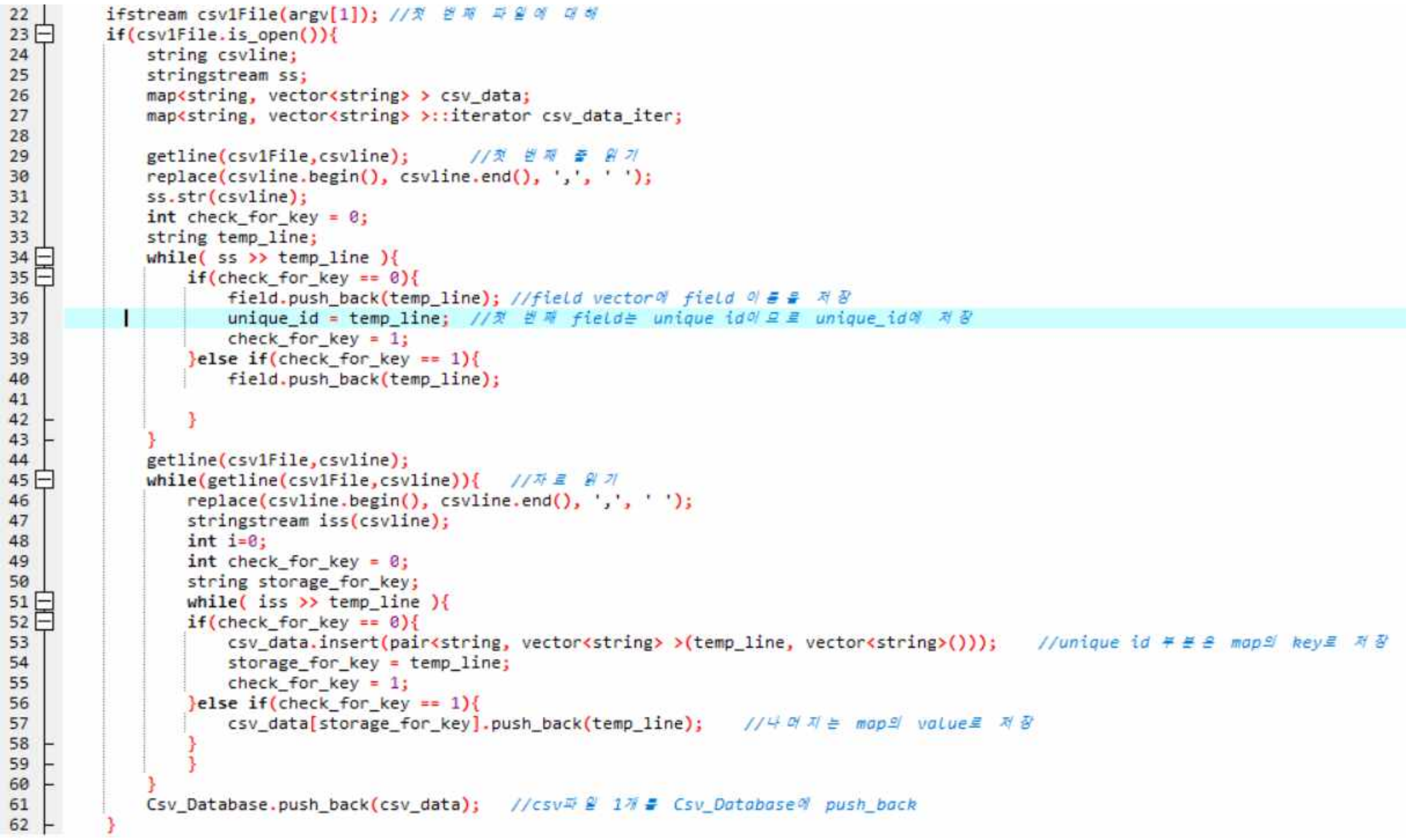
```
void SVG::print(FILE *html_out_){
    fprintf(html_out_,"<%s",tag_name.c_str());
    for(attribute_iter = attribute.begin(); attribute_iter != attribute.end(); attribute_iter++){
        fprintf(html_out_," %s = \"%s\" ",attribute_iter -> first.c_str(), attribute_iter -> second.c_str());       //attribute print
    }
    fprintf(html_out_,">");
    for(int i=0; i<svg_child.size(); i++){      //자식에 대해 print수행
        svg_child[i] -> print(html_out_);
    }
    fprintf(html_out_,"</%s>",tag_name.c_str(),if_selected);
}
```

그림 17 print

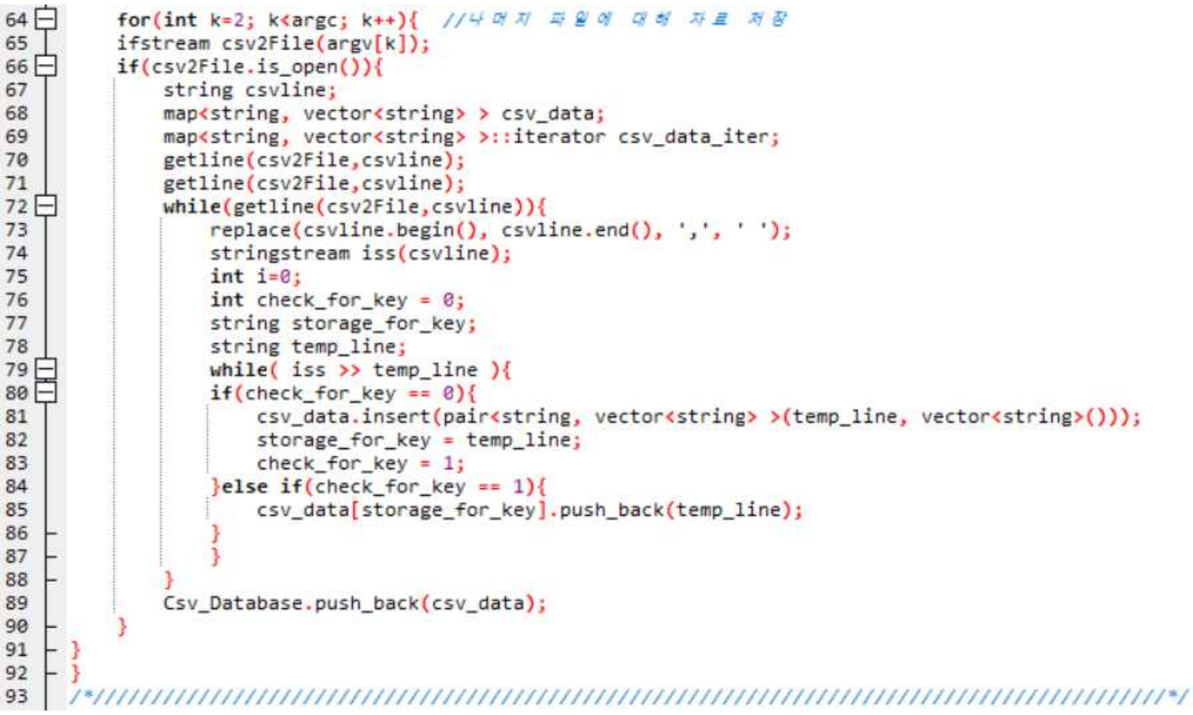
3)interpreter.cpp



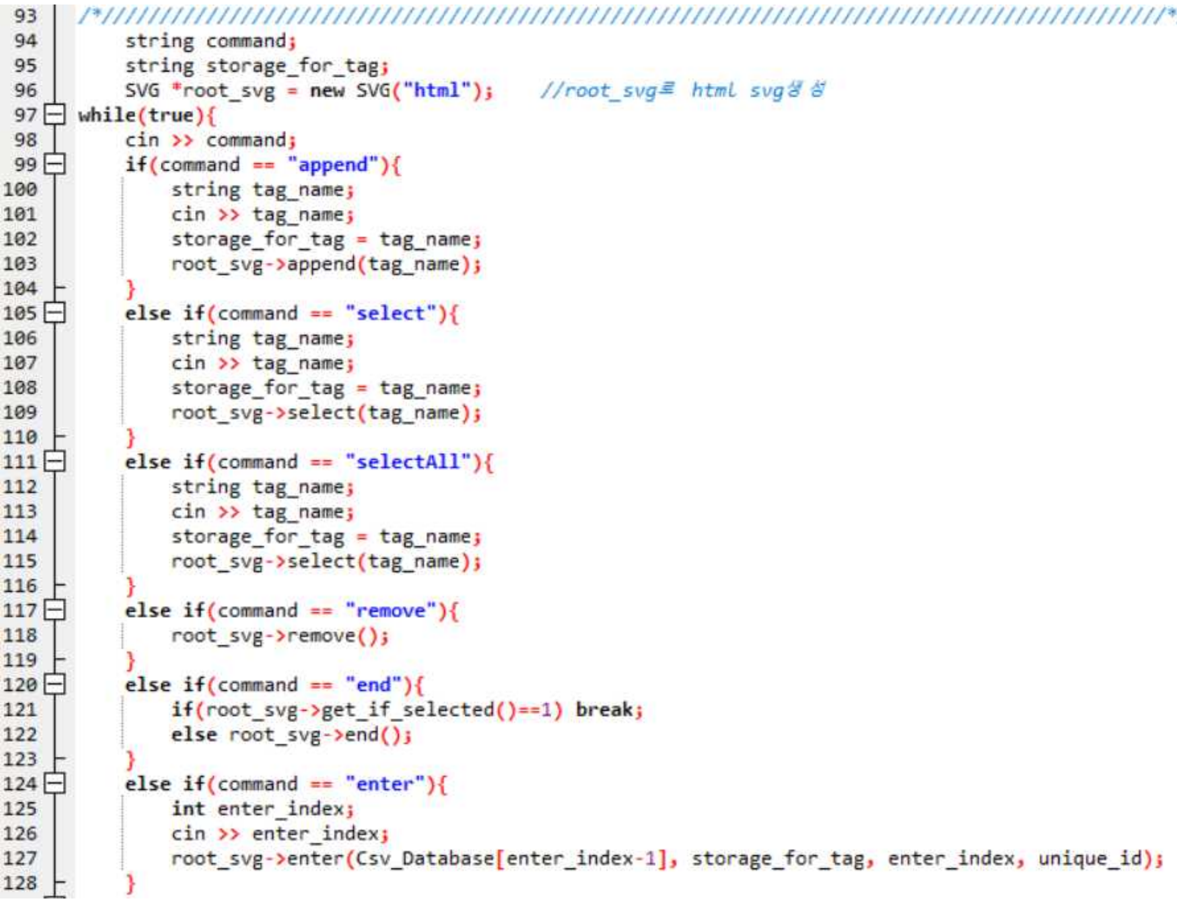
```cpp
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include <fstream>
#include <sstream>
#include <algorithm>

using namespace std;
#include "SVG.h"
#include "SVG.cpp"

int main(int argc, char** argv){
    int csv_index = 0;
    int data_number=0;
    vector<map<string, vector<string> > > Csv_Database;
    vector<string> field;
    string unique_id;

    if(argc>1){ //파일이 1개 이상 입력 되었다면

```



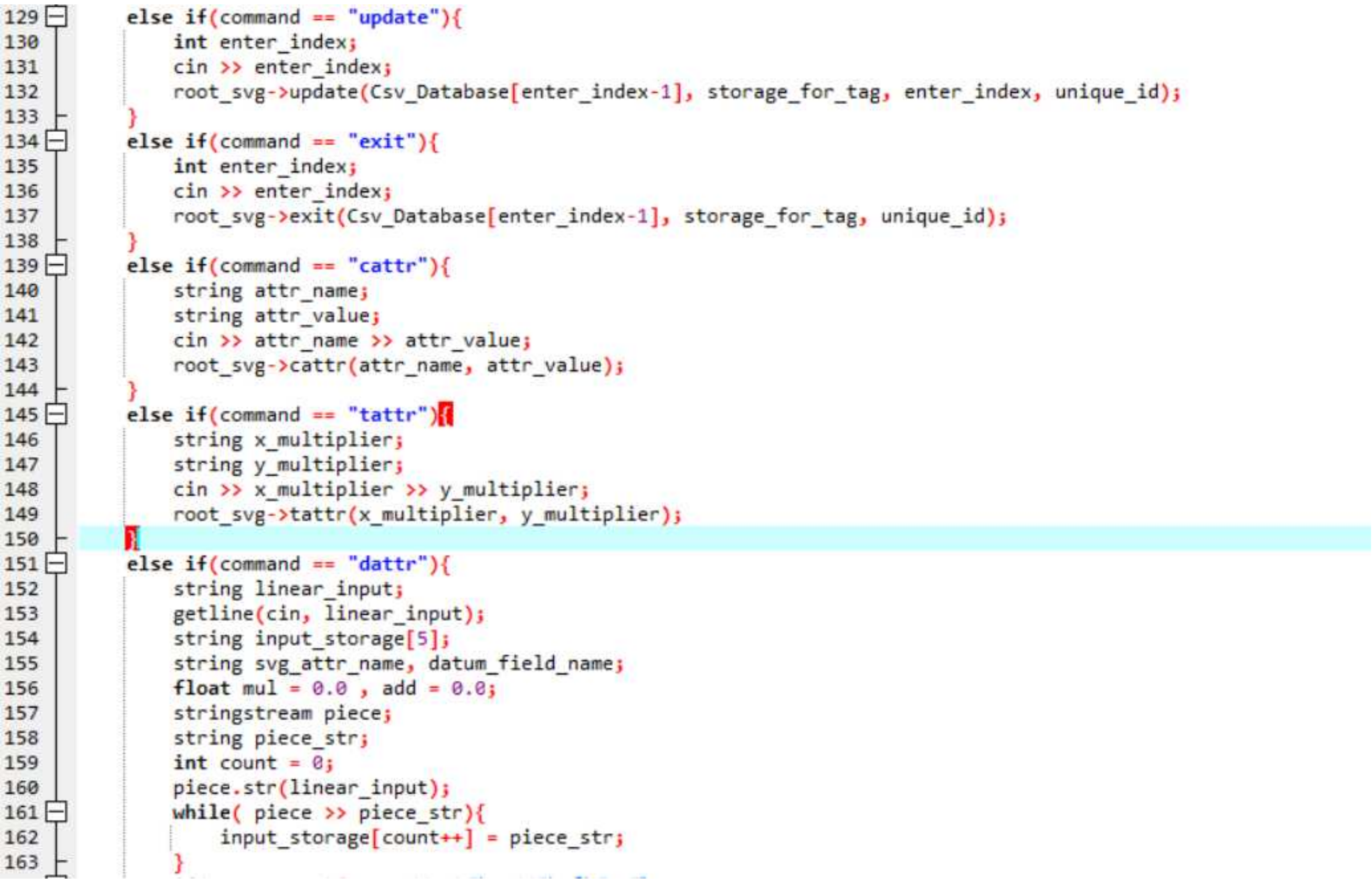
```cpp
    ifstream csv1File(argv[1]); //첫 번째 파일에 대해
    if(csv1File.is_open()){
        string csvline;
        stringstream ss;
        map<string, vector<string> > csv_data;
        map<string, vector<string> >::iterator csv_data_iter;

        getline(csv1File,csvline);      //첫 번째 줄 읽기
        replace(csvline.begin(), csvline.end(), ',', ' ');
        ss.str(csvline);
        int check_for_key = 0;
        string temp_line;
        while( ss >> temp_line ){
            if(check_for_key == 0){
                field.push_back(temp_line); //field vector에 field 이름을 저장
                unique_id = temp_line;  //첫 번째 field는 unique id이므로 unique_id에 저장
                check_for_key = 1;
            }else if(check_for_key == 1){
                field.push_back(temp_line);

            }
        }
        getline(csv1File,csvline);
        while(getline(csv1File,csvline)){   //자료 읽기
            replace(csvline.begin(), csvline.end(), ',', ' ');
            stringstream iss(csvline);
            int i=0;
            int check_for_key = 0;
            string storage_for_key;
            while( iss >> temp_line ){
            if(check_for_key == 0){
                csv_data.insert(pair<string, vector<string> >(temp_line, vector<string>()));    //unique id 부분은 map의 key로 저장
                storage_for_key = temp_line;
                check_for_key = 1;
            }else if(check_for_key == 1){
                csv_data[storage_for_key].push_back(temp_line);    //나머지는 map의 value로 저장
            }
            }
        }
        Csv_Database.push_back(csv_data);   //csv파일 1개를 Csv_Database에 push_back
    }
```



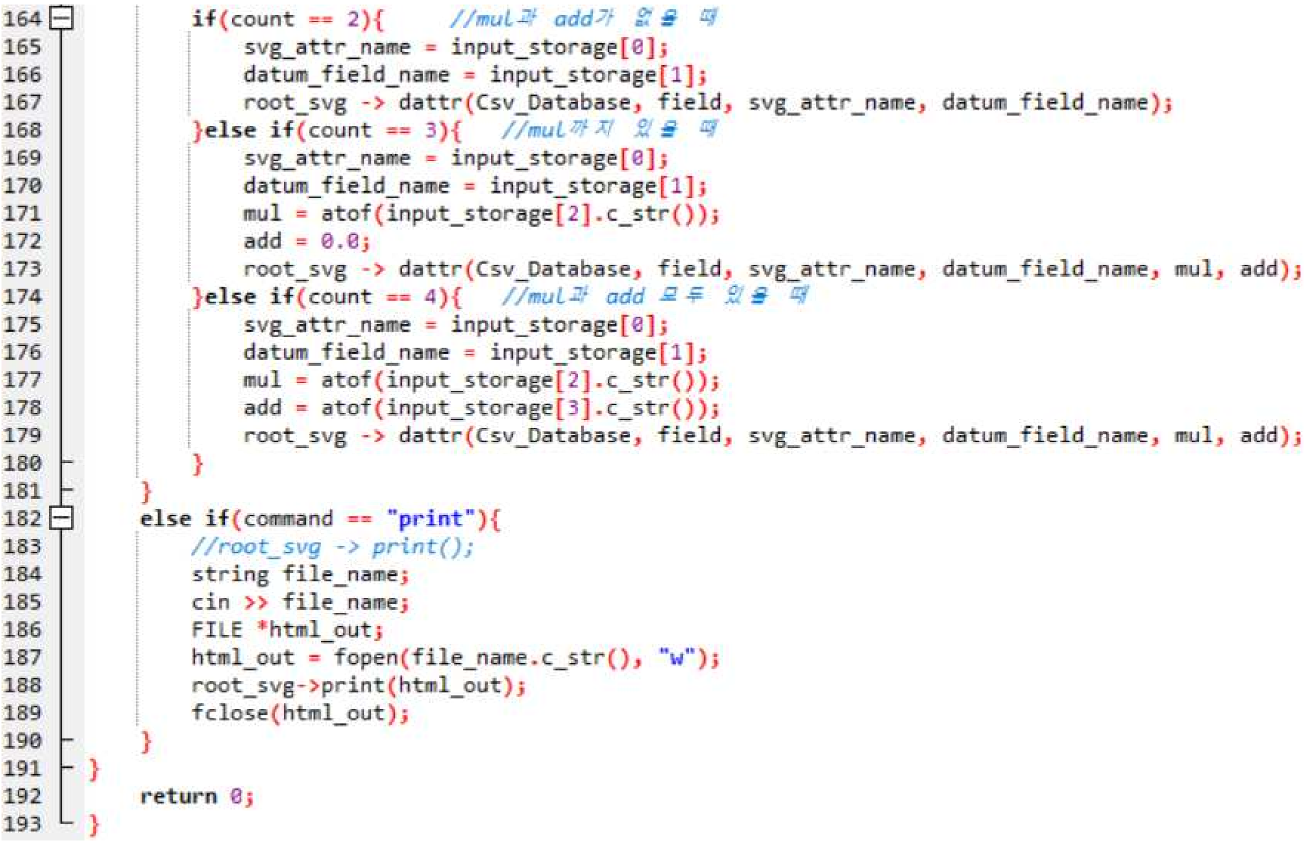
```cpp
    for(int k=2; k<argc; k++){  //나머지 파일에 대해 자료 저장
    ifstream csv2File(argv[k]);
    if(csv2File.is_open()){
        string csvline;
        map<string, vector<string> > csv_data;
        map<string, vector<string> >::iterator csv_data_iter;
        getline(csv2File,csvline);
        getline(csv2File,csvline);
        while(getline(csv2File,csvline)){
            replace(csvline.begin(), csvline.end(), ',', ' ');
            stringstream iss(csvline);
            int i=0;
            int check_for_key = 0;
            string storage_for_key;
            string temp_line;
            while( iss >> temp_line ){
            if(check_for_key == 0){
                csv_data.insert(pair<string, vector<string> >(temp_line, vector<string>()));
                storage_for_key = temp_line;
                check_for_key = 1;
            }else if(check_for_key == 1){
                csv_data[storage_for_key].push_back(temp_line);
            }
            }
        }
        Csv_Database.push_back(csv_data);
    }
}
}
/*/////////////////////////////////////////////////////////////////////////////////////////*/
```



```cpp
/*/////////////////////////////////////////////////////////////////////////////////////////*/
    string command;
    string storage_for_tag;
    SVG *root_svg = new SVG("html");    //root_svg를 html svg생성
while(true){
    cin >> command;
    if(command == "append"){
        string tag_name;
        cin >> tag_name;
        storage_for_tag = tag_name;
        root_svg->append(tag_name);
    }
    else if(command == "select"){
        string tag_name;
        cin >> tag_name;
        storage_for_tag = tag_name;
        root_svg->select(tag_name);
    }
    else if(command == "selectAll"){
        string tag_name;
        cin >> tag_name;
        storage_for_tag = tag_name;
        root_svg->select(tag_name);
    }
    else if(command == "remove"){
        root_svg->remove();
    }
    else if(command == "end"){
        if(root_svg->get_if_selected()==1) break;
        else root_svg->end();
    }
    else if(command == "enter"){
        int enter_index;
        cin >> enter_index;
        root_svg->enter(Csv_Database[enter_index-1], storage_for_tag, enter_index, unique_id);
    }
```



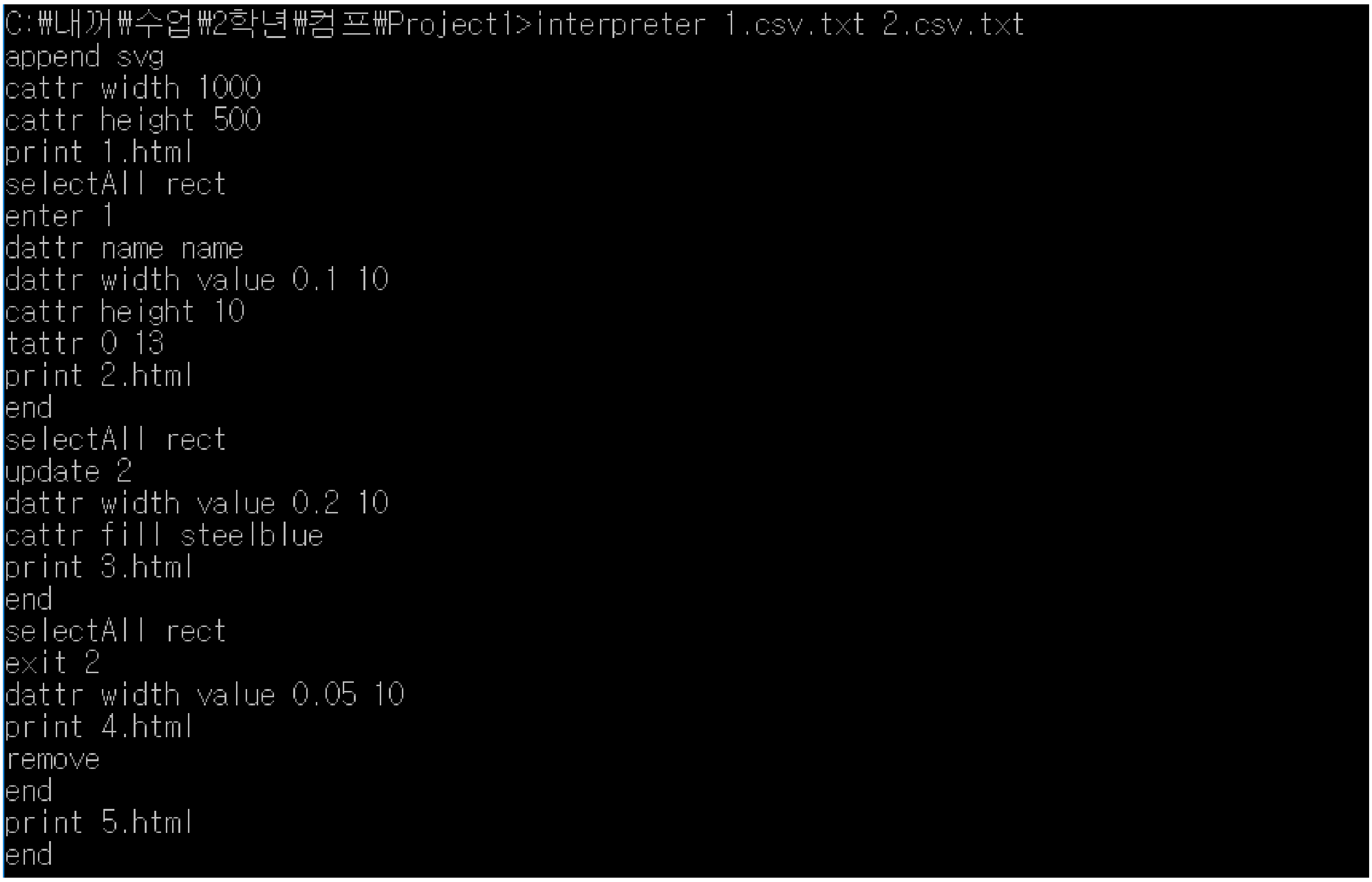
```cpp
    else if(command == "update"){
        int enter_index;
        cin >> enter_index;
        root_svg->update(Csv_Database[enter_index-1], storage_for_tag, enter_index, unique_id);
    }
    else if(command == "exit"){
        int enter_index;
        cin >> enter_index;
        root_svg->exit(Csv_Database[enter_index-1], storage_for_tag, unique_id);
    }
    else if(command == "cattr"){
        string attr_name;
        string attr_value;
        cin >> attr_name >> attr_value;
        root_svg->cattr(attr_name, attr_value);
    }
    else if(command == "tattr"){
        string x_multiplier;
        string y_multiplier;
        cin >> x_multiplier >> y_multiplier;
        root_svg->tattr(x_multiplier, y_multiplier);
    }
    else if(command == "dattr"){
        string linear_input;
        getline(cin, linear_input);
        string input_storage[5];
        string svg_attr_name, datum_field_name;
        float mul = 0.0 , add = 0.0;
        stringstream piece;
        string piece_str;
        int count = 0;
        piece.str(linear_input);
        while( piece >> piece_str){
            input_storage[count++] = piece_str;
        }
```



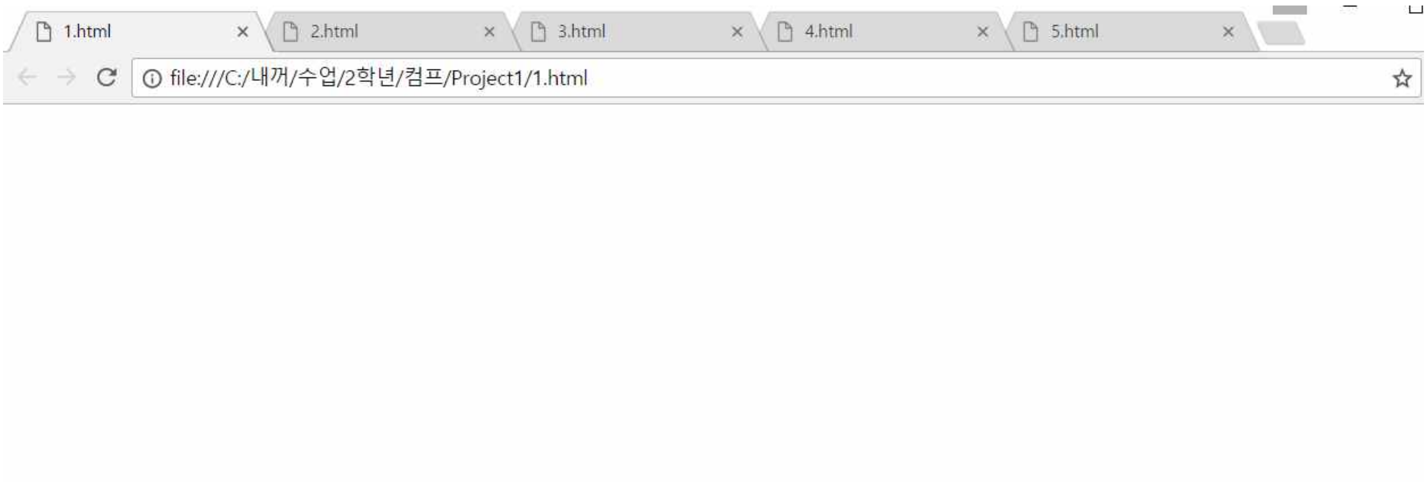
```cpp
         if(count == 2){      //mul과 add가 없을 때
             svg_attr_name = input_storage[0];
             datum_field_name = input_storage[1];
             root_svg -> dattr(Csv_Database, field, svg_attr_name, datum_field_name);
         }else if(count == 3){   //mul까지 있을 때
             svg_attr_name = input_storage[0];
             datum_field_name = input_storage[1];
             mul = atof(input_storage[2].c_str());
             add = 0.0;
             root_svg -> dattr(Csv_Database, field, svg_attr_name, datum_field_name, mul, add);
         }else if(count == 4){   //mul과 add 모두 있을 때
             svg_attr_name = input_storage[0];
             datum_field_name = input_storage[1];
             mul = atof(input_storage[2].c_str());
             add = atof(input_storage[3].c_str());
             root_svg -> dattr(Csv_Database, field, svg_attr_name, datum_field_name, mul, add);
         }
     }
     else if(command == "print"){
         //root_svg -> print();
         string file_name;
         cin >> file_name;
         FILE *html_out;
         html_out = fopen(file_name.c_str(), "w");
         root_svg->print(html_out);
         fclose(html_out);
     }
 }

     return 0;
 }
```

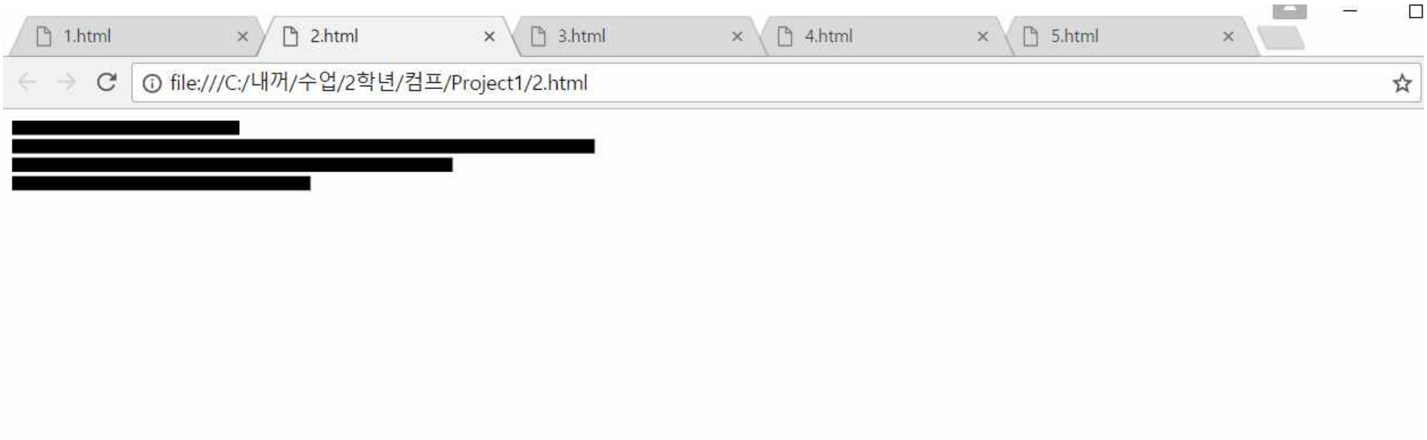
2. simulation & result

1)예제 1번



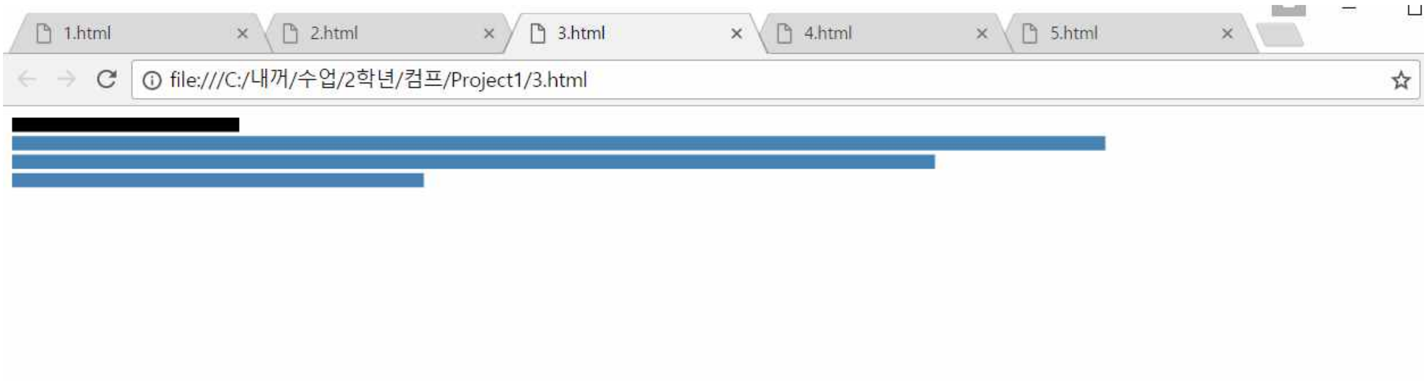
```
C:₩내꺼₩수업₩2학년₩컴프₩Project1>interpreter 1.csv.txt 2.csv.txt
append svg
cattr width 1000
cattr height 500
print 1.html
selectAll rect
enter 1
dattr name name
dattr width value 0.1 10
cattr height 10
tattr 0 13
print 2.html
end
selectAll rect
update 2
dattr width value 0.2 10
cattr fill steelblue
print 3.html
end
selectAll rect
exit 2
dattr width value 0.05 10
print 4.html
remove
end
print 5.html
end
```





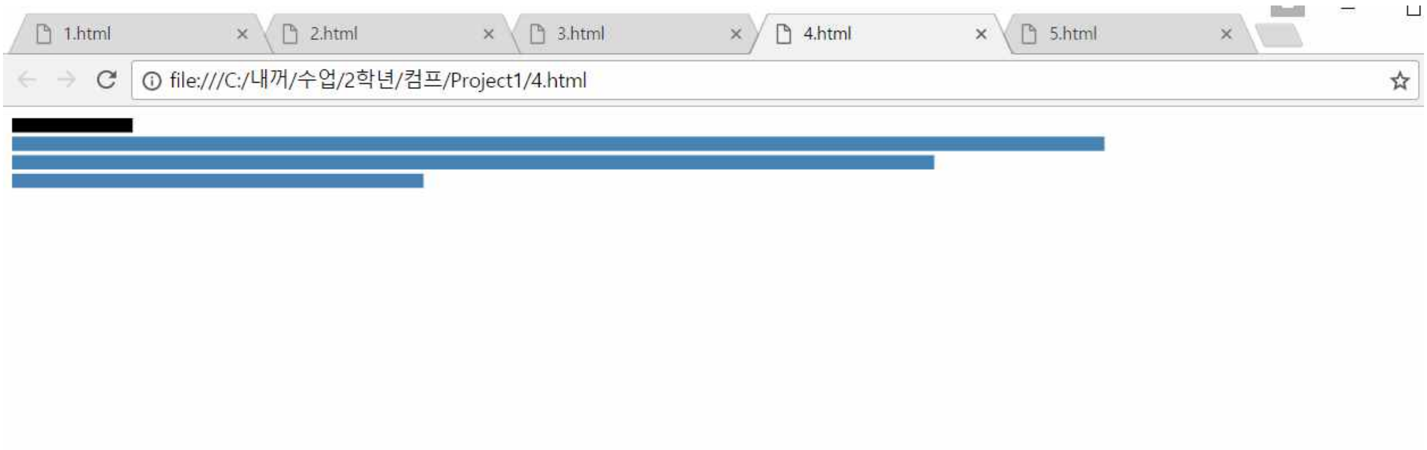
file:///C:/내꺼/수업/2학년/컴프/Project1/2.html

[redacted]



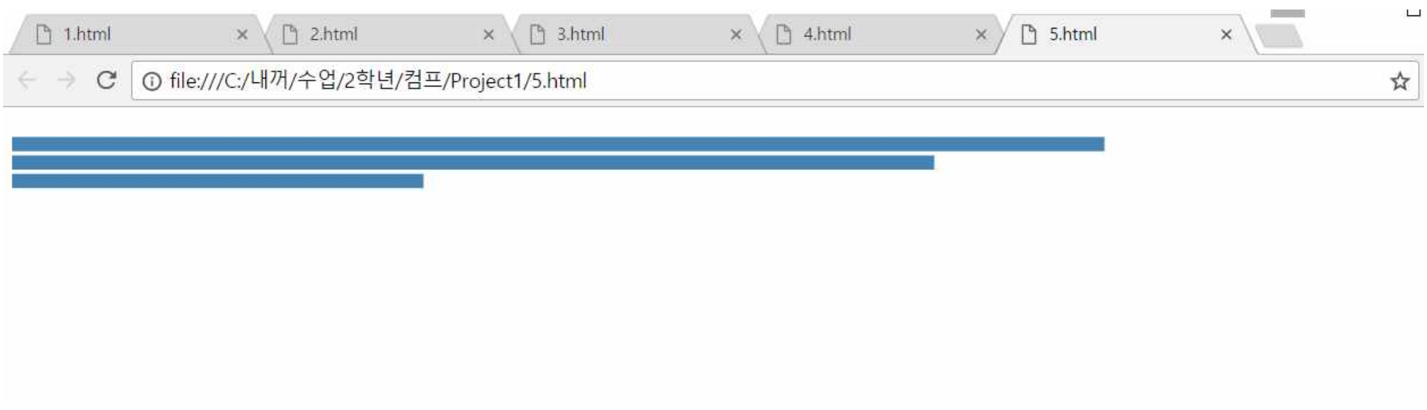
file:///C:/내꺼/수업/2학년/컴프/Project1/3.html

[redacted]

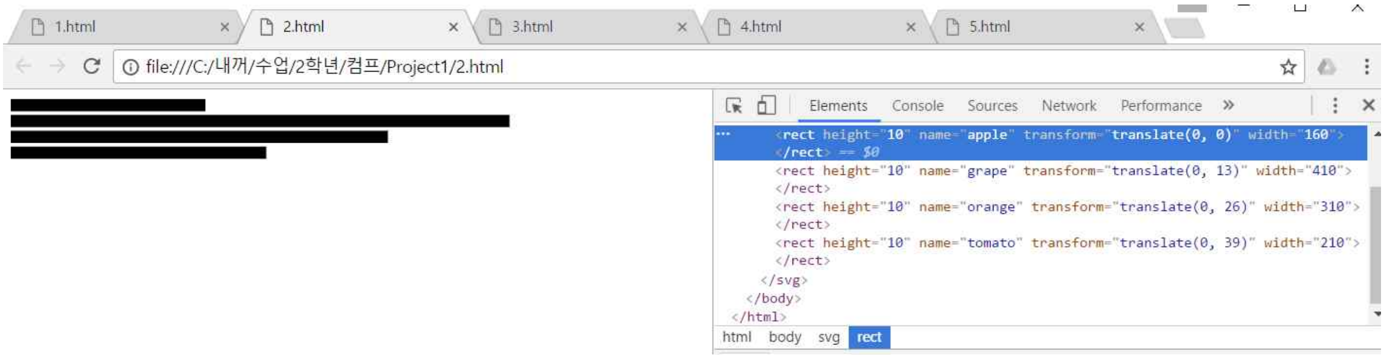


file:///C:/내꺼/수업/2학년/컴프/Project1/4.html

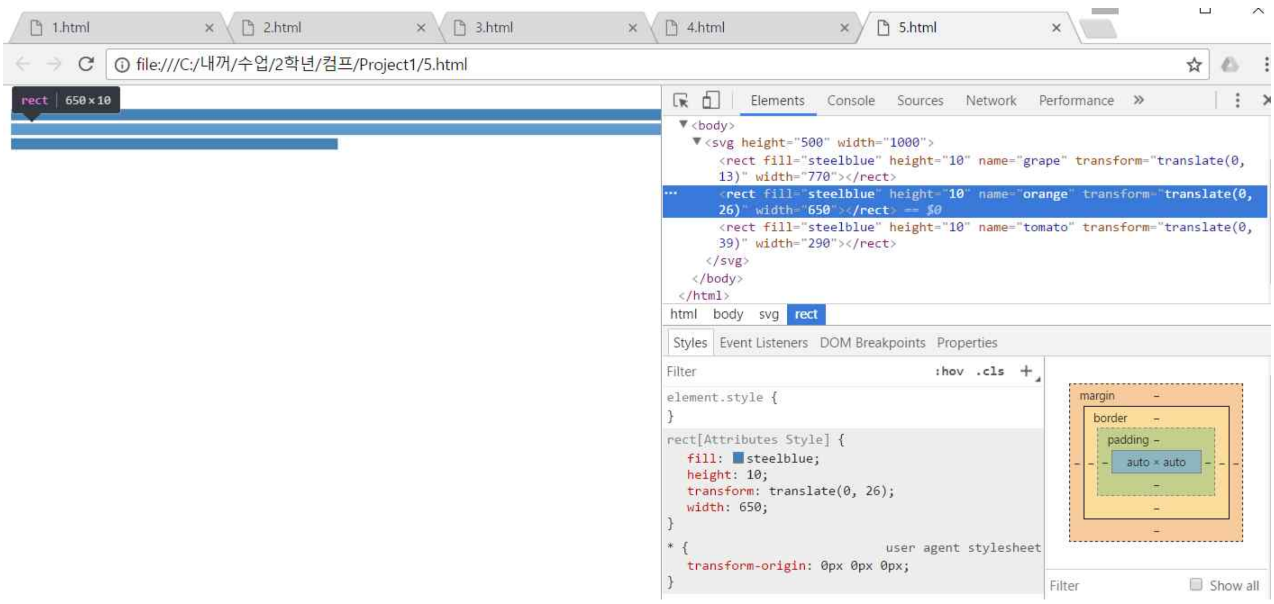
[redacted]



file:///C:/내꺼/수업/2학년/컴프/Project1/5.html



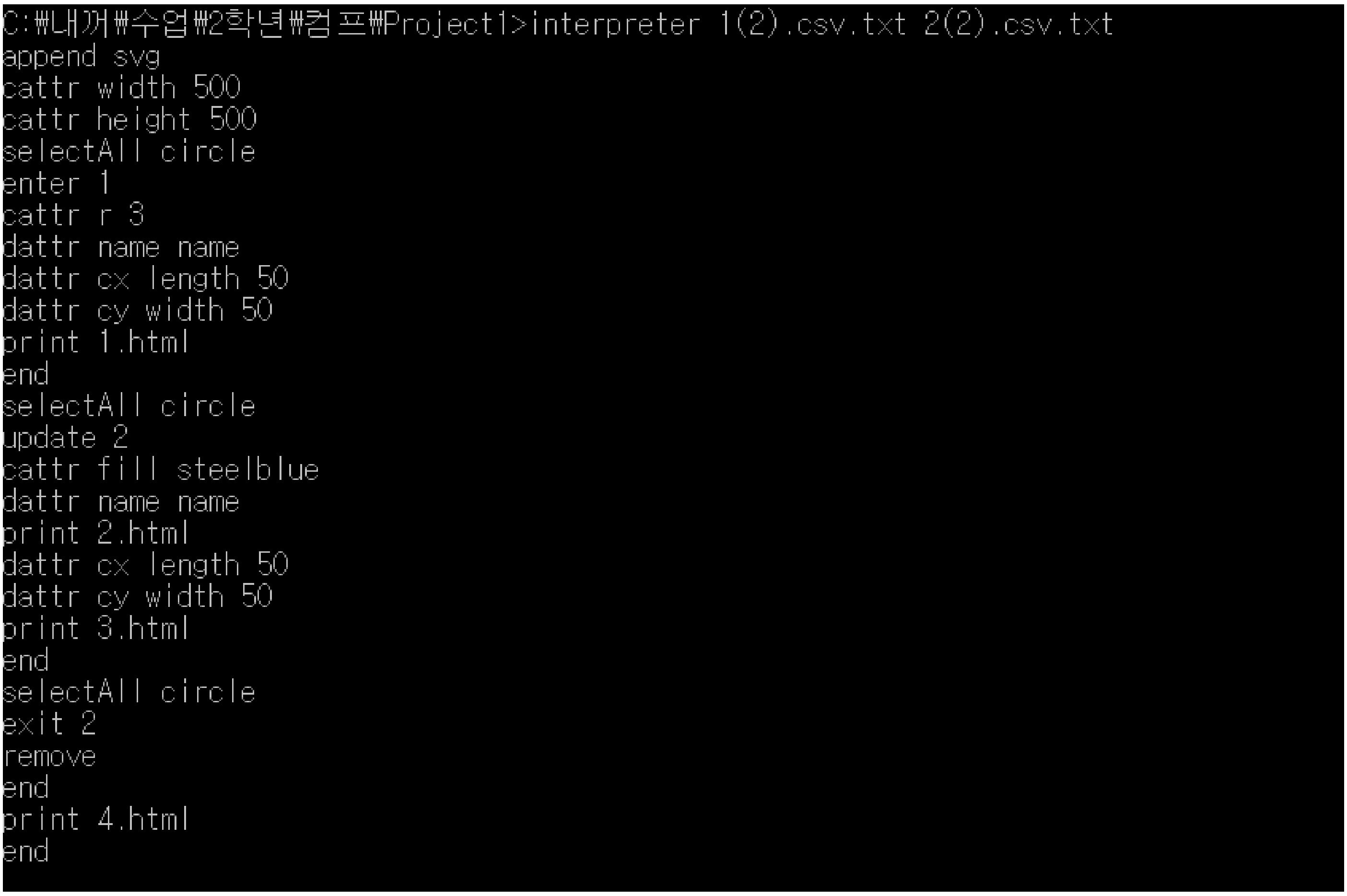
```
<rect height="10" name="apple" transform="translate(0, 0)" width="160">
</rect> == $0
<rect height="10" name="grape" transform="translate(0, 13)" width="410">
</rect>
<rect height="10" name="orange" transform="translate(0, 26)" width="310">
</rect>
<rect height="10" name="tomato" transform="translate(0, 39)" width="210">
</rect>
</svg>
</body>
</html>
```



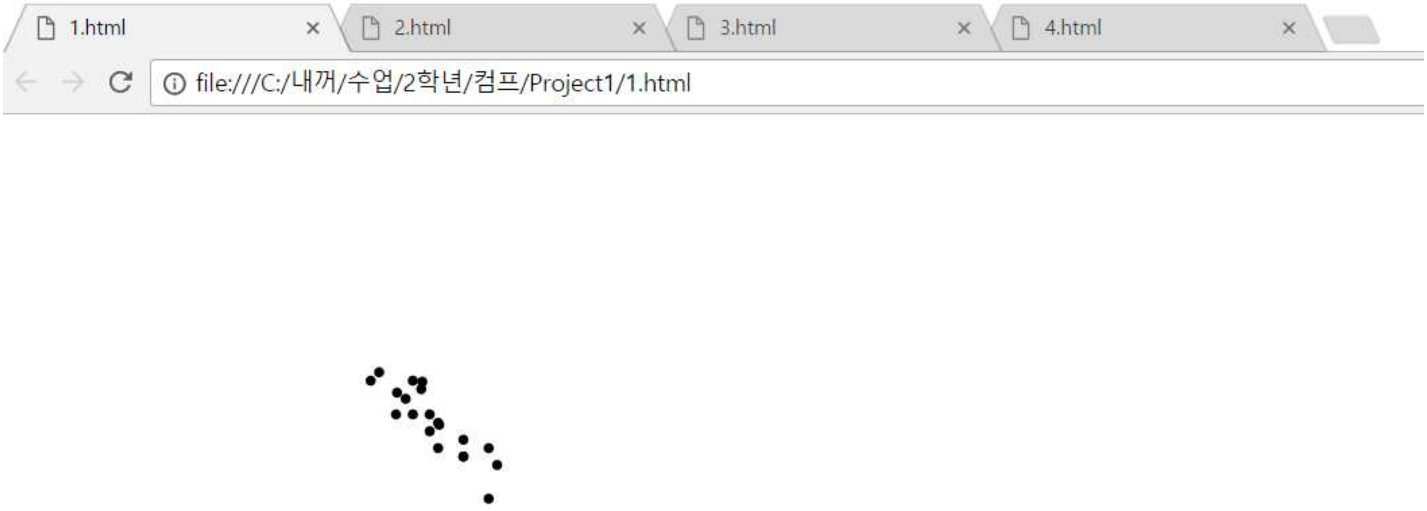
```
▼<body>
 ▼<svg height="500" width="1000">
    <rect fill="steelblue" height="10" name="grape" transform="translate(0,
    13)" width="770"></rect>
    <rect fill="steelblue" height="10" name="orange" transform="translate(0,
    26)" width="650"></rect> == $0
    <rect fill="steelblue" height="10" name="tomato" transform="translate(0,
    39)" width="290"></rect>
  </svg>
 </body>
</html>
```

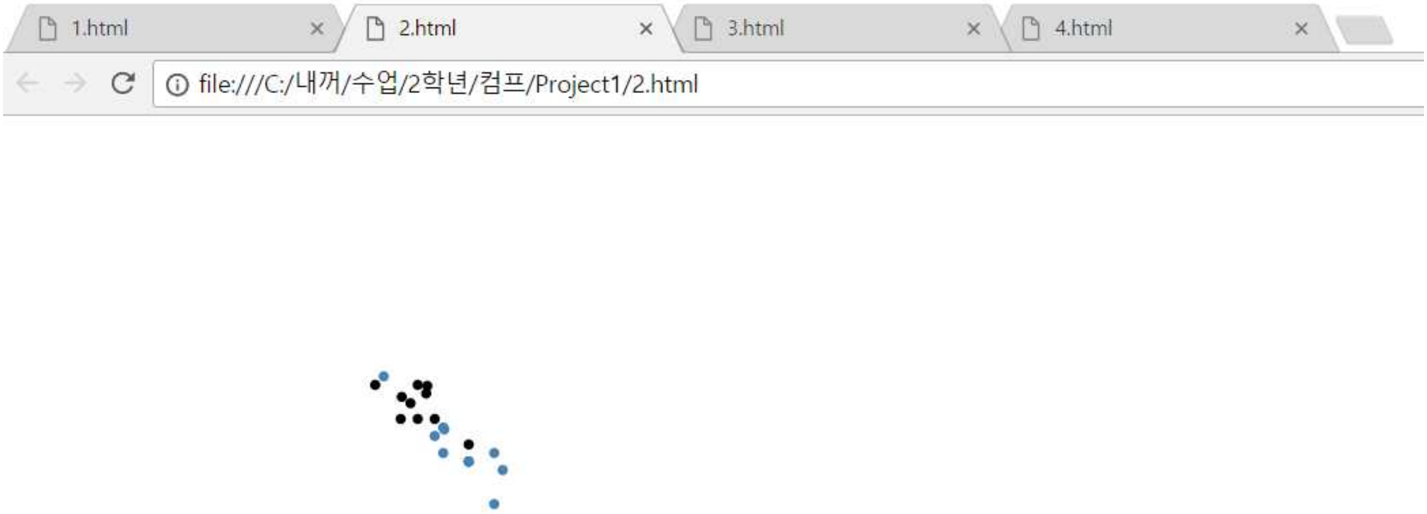
```
element.style {
}
rect[Attributes Style] {
  fill: steelblue;
  height: 10;
  transform: translate(0, 26);
  width: 650;
}
* {                         user agent stylesheet
  transform-origin: 0px 0px 0px;
}
```

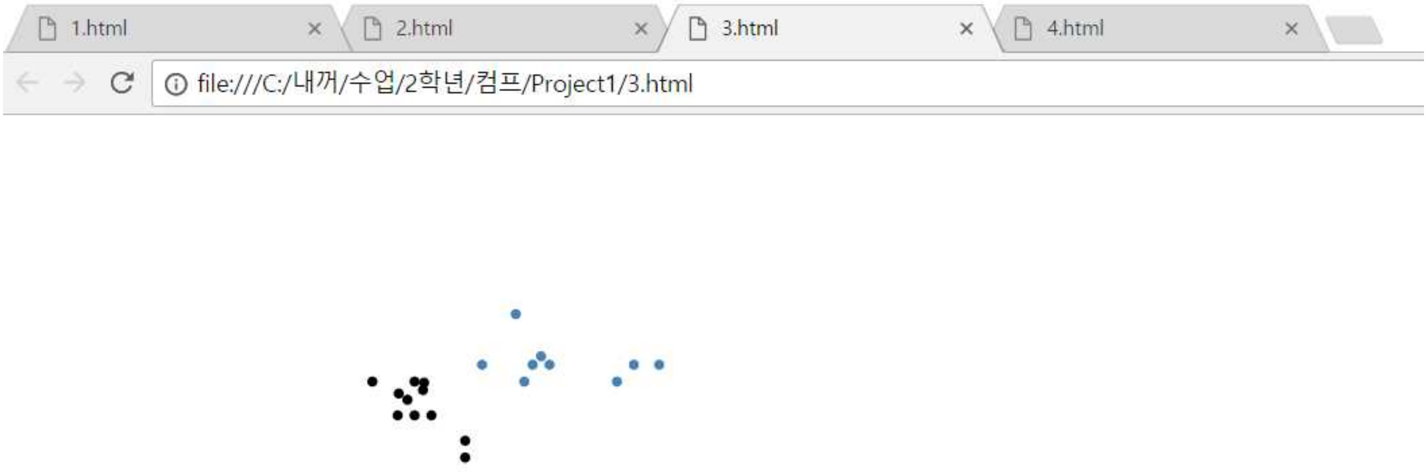
2)예제 2번



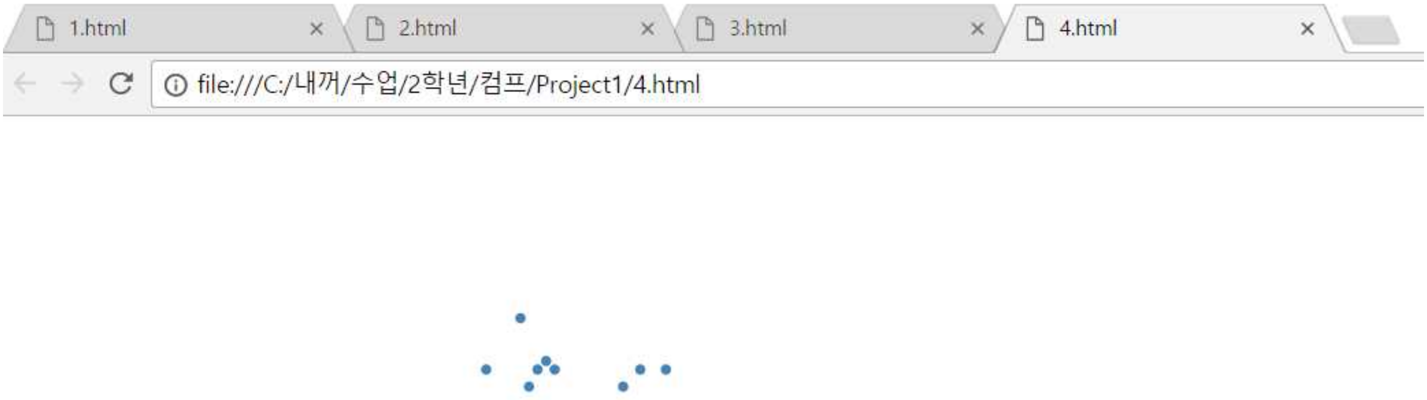
```
C:₩내꺼₩수업₩2학년₩컴프₩Project1>interpreter 1(2).csv.txt 2(2).csv.txt
append svg
cattr width 500
cattr height 500
selectAll circle
enter 1
cattr r 3
dattr name name
dattr cx length 50
dattr cy width 50
print 1.html
end
selectAll circle
update 2
cattr fill steelblue
dattr name name
print 2.html
dattr cx length 50
dattr cy width 50
print 3.html
end
selectAll circle
exit 2
remove
end
print 4.html
end
```


1.html
2.html
3.html
4.html
file:///C:/내꺼/수업/2학년/컴프/Project1/1.html


1.html
2.html
3.html
4.html
file:///C:/내꺼/수업/2학년/컴프/Project1/2.html



file:///C:/내꺼/수업/2학년/컴프/Project1/3.html



file:///C:/내꺼/수업/2학년/컴프/Project1/4.html