VG101 — Introduction to Computer and Programming

Lab 6

Manuel — UM-JI (Fall 2017)

Goals of the lab

- Understand attributes and methods
- Implement simple classes
- Connect C and C++

Ex. 1 — *From C to C++*

Write the C++ counterpart class of the following C structure and functions.

From C to C++

```c
#include <stdio.h>

static const char GRADES[] = {'F','F','F','F','F','F','D','C','B','A','A'};

typedef struct _Grade {
  char ltr;
  int prct;
} Grade;

void GradePrct(Grade *grade, int prct) {
  grade->prct = prct;
  grade->ltr = GRADES[prct / 10];
}

void GradeLtr(Grade *grade, char ltr) {
  grade->ltr = ltr;
  grade->prct = 100 - (ltr - 'A') * 10 - 5;
}

void printGrade(Grade *grade) {
  printf("Grade: %d -> %c\n", grade->prct, grade->ltr);
}

int main() {
  Grade g;
  int prct;

  printf("Input two space seprated grades (1st in %%, 2nd in letter): ");
  scanf("%d", &prct);
  scanf("\n");

  GradePrct(&g, prct);
  printGrade(&g);

  GradeLtr(&g, getchar());
  printGrade(&g);

  return 0;
}
```

Ex. 2 — *Class specification*

For an adventure game, a class allowing the generation of different investigators needs to be specified. A list of a few basic characteristics is provided:

- name
- sanity
- search
- stamina
- luck
- speed
- hide
- lore
- listen
- dodge
- persuade
- focus
- escape
- fight
- job
- run
- possession
- visit
- climb
- home

1. Separate the methods from the attributes and provide potentially missing attributes
2. Write the class `Investigator`, that defines the methods and attributes of an investigator. In particular define whether a method or an attribute should be public or private.
   Note that it is not asked to implement the class, but only to define it.

Ex. 3 — *Class implementation*

Following the definition of the `circle` class in chapter 13, write and implement the following simple classes.

- Triangle
- Rectangle
- Parallelogram
- Trapezium