

Poniższe projekty dotyczą tworzenia klasy danych abstrakcyjnego typu. Każda z napisanych metod **musi** zostać przetestowana w programie głównym napisanym w pliku `main.cpp`.

Oczywiście wszystkie projekty powinny być napisane w języku C++. Projekt powinien dać się skompilować i uruchomić na komputerach w pracowni 1203. Przed oddaniem projektu należy wysłać go e-mailem.

Proszę, aby wszystkie projekty w miarę możliwości pisać zgodnie z konwencją, o której mówiłem na zajęciach. Bardzo skrócony opis tych konwencji (dotyczący Javy, ale z powrotem możliwy do wykorzystania w C++) można znaleźć (niestety po angielsku) na stronie firmy Sun.

Oceniane będą (kolejność nieprzypadkowa)

1. SAMODZIELNOŚĆ WYKONANIA PROJEKTU
2. Poprawność działania i zgodność z zadaniem projektowym
3. Const-poprawność kodu,
4. Obsługa sytuacji wyjątkowych
5. Styl kodowania
6. Efektywność kodu

Projekt 1 *Napisać klasę reprezentującą elementy z ciała \mathbb{Z}_{13} . Należy zaimplementować*

- *wszystkie operatory logiczne,*
- *wszystkie operatory arytmetyczne i jednoargumentowe,*
- *wszystkie operatory przypisania,*
- *powyższe operatory powinny działać z typami prostymi,*
- *operatory inkrementacji i dekrementacji,*
- *wszystkie konieczne konstruktory i destruktor,*
- *współpracę z obiektami `cin` oraz `cout` biblioteki standardowej,*

Projekt 2 *Napisać klasę reprezentującą elementy z ciała \mathbb{C} . Należy zaimplementować*

- *wszystkie operatory logiczne (porównanie modułów liczb),*
- *wszystkie operatory arytmetyczne i jednoargumentowe,*
- *wszystkie operatory przypisania,*
- *powyższe operatory powinny działać z typami prostymi,*
- *metody zwracające moduł, sprzężenie i części rzeczywiste i urojone liczby,*
- *operatory inkrementacji i dekrementacji,*
- *wszystkie konieczne konstruktory i destruktor,*
- *współpracę z obiektami `cin` oraz `cout` biblioteki standardowej,*